Coordination of Systems through Numeraires

Paulo Garrido
Associate Professor
University of Minho
School of Engineering
Portugal

Visiting Professor
LIAAD INESCTEC

A paradigm for coordination of systems is proposed based on a controller with enough power to induce coordination among systems with respect to some goal.

The power considered here is of organization dissolution or system deletion, conditional on the values of an exchangeable scalar criterion applied to the system by the controller. The criterion is called a numeraire because the paradigm was abstracted from economic systems based on money.

One describes a method for the computer representation of producing systems based on string rewrite rules. Then, one studies the regulation problem for an abstract example of a coordinated system through a numeraire. First, one considers to define and enforce a nominal operation point; second, one studies the effects of disturbances and what can the controller do to minimize them.

Although the example is simple, it shows instructive results and allows to consider several research lines.

Keywords: *systems coordination, coordination through numeraires, coordination as causation, string production rules, coordination of large systems, chartalism.*

1. INTRODUCTION

The purpose of this paper is to present concepts of a paradigm to understand, model, simulate and design a type of coordination among systems.

The paradigm is called coordination using numeraires. It was abstracted from economic systems that use money. The notion of money is itself abstracted in the concept of a numeraire.

Consider a supersystem ***S*** made up of systems, $S(i)$, $i = 1, 2, ..., I$, and a controller C. To attain a set of systemic goals G, *all* the $S(i)$ must coordinate, but they require action from C to actually coordinate.

One assumes that the controller has the following powers:

– It can *set* for all $S(i)$ a *numerical criterion* $N(i)$. For brevity, such numerical criterion is called a numeraire.

– It will repeatedly decrease along time the values of all $N(i)$.

– It will repeatedly increase along time the values of numeraires of all systems that demonstrate the realization of the goals G.

– It will delete any system whose numeraire becomes negative.

One also assumes that systems can exchange among them their products, goods or behaviors, against numeraire. This is necessary because one assumes that all systems are required for the realization of G while only some can demonstrate and receive numeraire for it.

Under this set of rules, we assume that *the systems will try to avoid deletion*, therefore they will take decisions and actions in order to keep their numeraires non-negative. The *only* way they can get this result is by demonstrating the realization of G for which the controller "pays".

One understands *coordination through a numeraire* as the existence in a supersystem ***S*** of a set of behavioral rules as described above.

The development of the paradigm has as scientific aim to extend control and coordination theory by formalizing the operation of economic systems that use money as a coordination inducing device.

To the best knowledge of this author, this was not made elsewhere as is done here in a control and coordination perspective.

Three areas of application for the paradigm in development can be envisaged:

– The coordination of *large systems*. Tumer and Knudson [1] note that examples of such systems include packet routing across a data network, control of an array of communication satellites, coordination of multiple robots and machines, or dynamic job scheduling across a computer grid.

– The mapping of production processes *across enterprises*, eventually to whole societies, because the dynamics resulting from monetized transactions can be integrated with production dynamics. Putnik *et al* [2] presented a model for a large production network where this type of analysis would fit well.

– The *analysis* of some classes of economic phenomena.

In presenting the concepts developed for the paradigm, we propose that coordination be understood as (computational) causation. Again, to the best knowledge of this author, this was not made elsewhere.

Received: October 2015, Accepted: November 2015
Correspondence to: Paulo Garrido
Departamento de Electrónica Industrial,
Universidade do Minho, Campus de Azurém,
4804-533 Guimarães, Portugal,
E-mail: pgarrido@dei.uminho.pt
doi:10.5937/fmet1504377G

In this paper, the dynamics of coordinated systems are represented by string production rules. A simple set of production rules is given to make models for systems inside a scheme of coordination through a numeraire.

An example of a small system is developed and the following problems are addressed:

– Definition of an operational setpoint of production.

– The implications of the controller strategy of keeping constant the numeraire values for setpoint definition and production disturbance rejection.

We found the reported concepts of the paradigm relatively straightforward to apply and promising in terms of further development and application.

1.1 Plan of the paper

Section 2 presents the assumed concept of control and coordination. Causation is defined through a computational property. Section 3 presents the concept of a supersystem $\boldsymbol{S}$ being able of coordination towards (flow) goals using a controller C using a numeraire.

Section 4 presents the representation of the coordinated systems as string producing rules acting on inputs described by pairs of vectors. Each rule specifies the production of an output string from an input vector of component strings and a vector of quantities for each component. Rules may be of necessary or possible operation and may produce or not output that can be used in making other strings.

Section 5 considers the definition of a regulation problem for an example system. First, a structure and a control strategy defining nominal flows are considered. Second, disturbances are considered.

In Section 6, we reflect on the results of the previous sections and consider future research lines.

2. THE ASSUMED CONCEPTS OF CONTROL AND COORDINATION

What does it mean for a system $\boldsymbol{S}$ to be the seat of a control process? For the purposes of this paper, one interprets control as the reduction or constraining of the space of possible behaviors. For an example, consider a plane that must fly from city A to city B, beginning at the moment when the pilot ignites the motors at A. It is clear that the admissible or *controlled* set of behaviors in terms of air traffic and passengers comfort is a *very tiny subset* of all the possible behaviors, with most of these ones being the plane crashing, going to another place than y, etc., etc.

Formally, we can put things as follows. Let the system $\boldsymbol{S}$ have a vector of variables $\boldsymbol{x}(t)$ with a possible behavioral space $\boldsymbol{X}(t)$. In the case of the plane, $\boldsymbol{x}(t)$ is a vector with six components and $\boldsymbol{X}(t)$ is the set of all possible behaviors of the plane since the motors are ignited at A. Set a criterion J to define a proper subdomain $\boldsymbol{X}_r(t)$ of $\boldsymbol{X}(t)$, specified through the behavior of a vector $\boldsymbol{x}_r(t)$. In the case of the plane, the criterion J says that the plane must fly according to air traffic regulations and passengers comfort, along a nominal path $\boldsymbol{x}_r(t)$. This specifies the constrained, restricted or controlled subdomain of admissible behaviors of the plane $\boldsymbol{X}_r(t)$ in flying from A to B.

If there is a subprocess C in $\boldsymbol{S}$ that changes $\boldsymbol{S}$ so that $\boldsymbol{x}(t)$ belongs to $\boldsymbol{X}_r(t)$, we say that C controls $\boldsymbol{S}$ with respect to the vector $\boldsymbol{x}_r(t)$. In the case of the plane, the pilots and all the control devices they use constitute the “subprocess”.

William Ashby considered implicitly such constraining in what he called the law of requisite variety for controllers [3], [4]. It says that for compensation of a disturbance to be possible the number of states of a controller must be equal or greater than the number of the states of the presumed disturbance generator. The strictness of control is given by some measure of admissible deviation from target along time defining the subset of admissible behaviors $X_r(t)$ above. For this scalar case, the smaller the fraction $X_r(t)$ / $X(t)$, the higher the potential information supplied by C to S.

Control has a cost and this can be measured by the entropy produced as proposed by George Saridis [5]. Actually, a control process reduces the entropy of the trajectories of the system. This entropy must be taken out, so a stricter control must have a higher entropy removal cost.

We consider now coordination. A coordination problem can appear in any vector of variables with more than one component. Consider a XY-motor system that must draw a circumference. For this to happen, the X motor must change velocity according to a cosine function of time while the Y motor must do the same according to a sine. The evolution of variables must be coordinated. A usual way to do this is to set an electronic controller that will generate the voltage functions for the motors.

In this case, coordination has a very simple interpretation, but the words *coordination* and *control* appear together in complex systems, in management, in distributed systems, in robotic and control systems. We need a concept of coordination that allows us to connect the two concepts. Malone and co-workers [6] proposed a “coordination theory” mainly aimed to application in a human organizational stance; it will not cover the simple example of the two motors above.

Let us assume that data from two scalar variables, $x(k)$ and $y(k)$, becomes available. Could one detect coordination in them? The answer explored here is to equate coordination with causal connection.

The concept of ‘a causes b’ is taken to mean that *to compute* $b(k)$, the value of variable b at instant k, one uses values of variable a at instants *up to* $k-1$ or *including* k with the restriction that including k does not make $b(k)$ depend on $b(k)$.

There are several possibilities for two variables x and y *being causally related*, see Figure 1. They may have no discernable causal relation or connection, as in a). They may have no discernable causal relation between the two, but they may be both caused by a third variable u, as in b). It may be the case that x causes y either directly or indirectly but y does not cause x, as in c). The symmetrical situation to c) is depicted in d). Finally, it may be the case that x causes y either directly or indirectly and y does cause x either directly or indirectly making a causal loop, as in e).

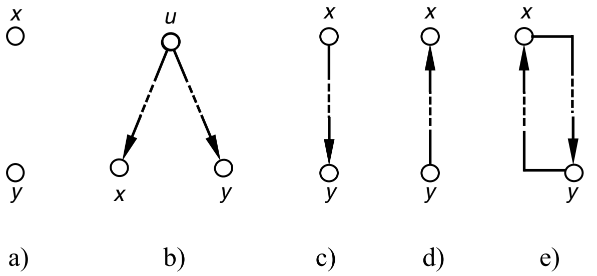

Figure 1. Possible causality relations between two variables

We note that defining coordination in this sense makes that *two coordinated variables present inherent control* because the causal connection reduces the behavioral space of both.

With this definition, we can say that many physical variables are coordinated; this makes sense, as "physical laws" can be expressed as programs to compute variables. This type of coordination is inherent in the fabric of the physical world and it has the curious property of being *inescapable*.

Let us note that two variables that are not coordinated can become coordinated and vice-versa. For complex systems, coordination or causal connection is not a property fixed in time. Moreover, coordination can have several consequences or effects. We are interested here in effects that are *flows* of products, with the understanding that behaviors can be considered products.

3. COORDINATION TOWARDS GOALS INDUCED THROUGH NUMERAIRES

Assume that it is possible for a supersystem $\boldsymbol{S}$, made up of a set of systems $S(i)$ and a controller C, to produce items in a set B at flows specified in a vector Y, along time periods indexed by a variable k. If $\boldsymbol{S}$ does produce the items at the specified flows in period k, we will say that $\boldsymbol{S}$ realizes the goal $G(k) = (B(k), Y(k))$.

Autonomous coordination can be divided in two types: with or without a controller C associated. The second one requires that the systems will take decisions through a completely decentralized process. If some sort of central decision exists, we can say that it is implemented by a controller C.

Conceivably, there are many possibilities for a set of systems to coordinate through the action of a controller.

We consider here coordination schemes that rest on controllers having a power of coercion over the coordinated systems and use the power to impose a criterion value to each system that we call a *numeraire*.

A controller C induces coordination through a numeraire if it is able to impose a criterion value of the numeraire type for each producer and change it along time. We express the numeraire value of each system i at instant k as $N(i, k)$. We will say that a criterion is a numeraire when it happens that:

– A system $S(i)$ is dissolved if its criterion becomes negative.

– C diminishes the $N(i)$ values according to a decreasing or tax transaction; the rule to calculate the decreasing is knowable for all systems.

– The controller or any other system can increase the numeraire of a system $S(i)$ through an increasing or purchase transaction, where $S(i)$ delivers a product and receives in exchange an amount of numeraire. C will purchase a set of designated $C(G)$ products from the goal G.

The controller fixes a cycle and applies a negative or null impulse in the numeraire of each producer at the end of each cycle. This may be recognized in a social stance as a social power applying a *tax* that must be *paid* in a *particular money* or *currency* chosen by the power. The tax paid will be calculated by a rule knowable to all producers. Systems for which a time integral of the criterion becomes negative are deleted. Therefore, the producers under consideration here are, in real terms, either machines or organizations. These are assumed to act and adapt in order to keep or increase the positiveness of its numeraire, referred in a social stance as the *money an organization owns*.

Producers can increase the values of their numeraires, or *earn* numeraire, by coordinating to make available to the controller products in a subset $C(G)$ of the goal G chosen by the controller C[1]. For each item made available, the controller will apply a positive impulse to the numeraire value of the supplier system. At the beginning of each cycle, the controller sets $C(G)$, together with the amounts of numeraire it will credit in exchange, or the *prices* it will *pay* for them. This may be recognized in a social stance as a social power *purchasing* products at a given price and paying for them to organizations.

Producers can also increase the values of their numeraires by producing items that other systems want to purchase or to exchange for their numeraire. To get the positive impulses required to avoid deletion, systems must *sell* the products in $C(G)$ to the controller or any other product that other systems may want to *buy*.

4. COORDINATING SYSTEMS AS STRING PRODUCERS

In this paper, we consider systems that coordinate *to produce strings of symbols*. There are two reasons for considering systems that produce strings of symbols.

The first is that whatever can have a verbal encoding must be encoded with strings of symbols. Therefore, in studying machines that can produce strings of symbols we are virtually studying all the machines that we can study modulo an interpretation of symbols to a supersystem of a given nature.

The second is that one can always see the coordinating systems as computer systems because computation can always be interpreted as string processing. In this view, a system $S(i)$ entering coordination is represented by a device that receives input strings and supplies output strings. The relation

[1] Note that this formualtion is a bit more precise that the one given in the introductory section.

between input and output strings is described by a set $W(i)$ of *rewriting rules*. All the rules in the supersystem will make a set W.

The content and atomic operation of a rewriting rule, w, as conceived here, can be represented most generally by the expression:

$$w: A \to \eta b\,. \tag{1}$$

It means that if provided with input strings as specified by the pair of vectors A, an instance of the rule w will *delete* the strings in A and will *create* a number η of copies of string b. The number η can be rational but unless necessary, we will take η to be one. Following common use, we will call A the *antecedent* and b the *consequent*.

The antecedent is a pair of vectors:

$$A = \left(\boldsymbol{c}(b), \boldsymbol{m}\left(\boldsymbol{c}(b)\right)\right). \tag{2}$$

The components $c_i(b)$ of $\boldsymbol{c}(b)$ are strings. Each $m_i(c_i(b))$ in $\boldsymbol{m}(\boldsymbol{c}(b))$ is the number of copies of string $c_i(b)$ required for the atomic operation of synthesizing the minimal amount η of string b copies. Values of $\boldsymbol{m}(\boldsymbol{c}(b))$ can be rational numbers.

Each instance of a rule has an *output flow y* or number of copies produced in a time period k. The flow will be always an integer multiple of η. For a rule to operate at output flow y in period k, then it must be provided with *input flows* $\boldsymbol{u}(k) = (y/\eta)\,\boldsymbol{m}(\boldsymbol{c}(b))$ of component strings as specified in $\boldsymbol{c}$.

An instance of a rule in a particular system can work at different values of output flow $y(k)$ with known minimum *ymin*, nominal *ynom*, and maximum *ymax*, flows. To these values, correspond the vectorial values of input flows $\boldsymbol{umin}(k)$, $\boldsymbol{unom}(k)$, $\boldsymbol{umax}(k)$.

Rules can be of necessary, $ymin > 0$, or possible, $ymin = 0$, operation. A rule is necessary if it happens that $y(k) < ymin$ for some k implies that any system hosting the rule *will be deleted*. A producer of strings can choose to make available the output product for reuse in other rules or not.

Beyond production of copies, producers must have transfer capabilities between them. In order to formally define transfer of strings we assume that producers have input and output storages for strings.

The basic structure and operation of a rule instance assumed is depicted in Figure 2.

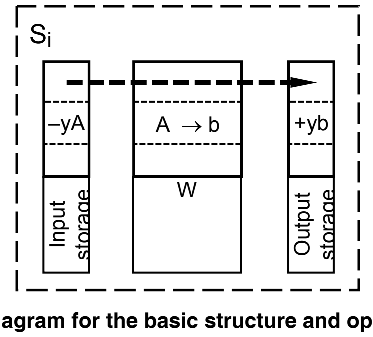

Figure 2. Diagram for the basic structure and operation of a rule instance

Let x be an integer and xA be defined as:

$$xA = x\boldsymbol{m}\left(\boldsymbol{c}(b)\right) \tag{3}$$

That is, xA is an integer multiple of the minimal numbers of different strings specified in A necessary to produce b. In one period k, a rule instance can create yb copies of consequent and sum them to those in the output storage if yA copies of the antecedent are deleted in the input storage of the producer. If producer $S(i)$ has yA copies at instant k–1, it can take the decision of producing yb copies available at instant k, provided that $yb \le ymax$. Said in other way, an output flow $y(k) = yb$ requires a (vectorial) input flow $\boldsymbol{u}(k) = y\boldsymbol{m}(\boldsymbol{c}(b)) = yA$.

Let uA be a multiple of the minimal numbers of different strings specified in A necessary to produce b that are present in the input storage. Let ub be the number of copies of b present in the output storage. The updating equations for both are:

$$\begin{cases} uA(k) = uA(k-1) - yA(k) \\ ub(k) = ub(k-1) + yb(k) \end{cases} \tag{4}$$

System coordination requires production and transactions, i.e., exchanges of strings for amounts of numeraire.

Exchanges of strings for amounts of numeraire are actually two coupled transfers: one of strings, the other of numeraire. A transfer of numeraire from system i to system j is recorded for system i as $n(i,j)$ and it has a negative value as the numeraire "goes away" from i. The same transfer of numeraire is recorded for agent j as $n(j, i)$ and it has a positive value as it "goes to" j. Therefore, the two records of a numeraire transfer n satisfy:

$$n(i,j) = -n(j,i) \tag{5}$$

If mb copies of a string b are in the output storage of $S(i)$ and this announced them for sale at price $rb(k)$, then a buyer $S(j)$ can command the transfer of $sb(k)$ copies to its input storage. It can do this on the conditions that $sb(k) \le mb(k)$ and it transferred a payment amount $n(j,i,k) = -sb(k)\cdot rb(k)$ from $N(j)$ to $N(i)$.

The updating system of equations for an exchange transfer occurring in a period k is:

$$\begin{cases} mb(i,k) = mb(i,k-1) - sb(i,j,k) \\ ub(j,k) = ub(j,k-1) + sb(i,j,k) \\ n(i,j,k) = -n(j,i,k) = sb(i,j,k)rb(i,k) \\ N(i,k) = N(i,k-1) + n(i,j,k) \\ N(j,k) = N(j,k-1) + n(j,i,k) \end{cases} \tag{6}$$

To simplify analysis and algorithms a bit, one considers that possible rules function in a produce-and-transfer mode in one period. This may be represented changing the first equation in (6):

$$\begin{cases} uA(i,k) = uA(i,k-1) - sb(i,j,k) \\ ub(j,k) = ub(j,k-1) + sb(i,j,k) \\ n(i,j,k) = -n(j,i,k) = sb(i,j,k)rb(i,k) \\ N(i,k) = N(i,k-1) + n(i,j,k) \\ N(j,k) = N(j,k-1) + n(j,i,k) \end{cases} \tag{7}$$

The first equation in (7) records now the decrease in the input storage of the producer system.

In this scheme, there are two decisions left for producers: which quantities and prices to announce for the products they offer for sale and which products to buy from those offered for sale.

The decisions of the controller are which products to announce for purchase at which prices and which rule taxes to apply.

In the next section, we define a regulation problem for a simplified class of supersystems. As a regulation problem, the goal of the controller will be to establish, drive and keep the flows of strings near the nominal values along time under a given set of disturbances.

5. STUDYING A REGULATION PROBLEM FOR AN EXAMPLE STRING PRODUCING SYSTEM

Assume that under undisturbed conditions of operation one can expect that the nominal flows of rules in producers, $Y_{nom}(i)$ are constant.

Let the controller assume these values and set the flow goals in G as $Y_r(i, k) = Y_{nom}(i)$.

We will say that the controller faces a *regulation problem*, if for some i and some k one verifies that $Y(i, k) \neq Y_{nom}(i)$. The problem will be solved for some horizon h if $Y(i, k+h) = Y_{nom}(i)$.

5.1 Defining the system and nominal values

As an example, we will consider a structure where producers have one possible rule and two necessary rules. The necessary rules considered have the form:

$$w_n : X \to '' \tag{8}$$

The double quote symbol means the usual or *bottom* empty string. Rules of the form (8) operate as "infinite capacity sinks" for strings, as any number of strings in the antecedent will be made to disappear. Two simple rules will be assumed:

$$\begin{aligned} &w_{n1} : '\mathrm{e}' \to '' \\ &w_{n2} : '\mathrm{f}' \to '' \end{aligned} \tag{9}$$

If the rules are present in a system i then the system must consume *ymin* units of 'e' and 'f' strings to persist in time.

This means that the total capacities of the system for producing 'e' and 'f' strings must be equal or greater than the sum of the individual needs.

We will assume that the rule to produce 'e' is of the form:

$$w_{*} : * \to \eta b \tag{10}$$

The symbol * can be understood as a *top* empty string, because it acts as an "infinite capacity source": copies of any string can be produced by a rule with the format as in (10) without input strings.

We assume that the two productive rules for 'e' and 'f' are:

$$\begin{aligned} &w_e : * \to '\mathrm{e}' \\ &w_f : 2'\mathrm{e}' \to '\mathrm{f}' \end{aligned} \tag{11}$$

We also assume a rule that will produce the string that the controller will buy:

$$w_d : \{2'\mathrm{a}', 3'\mathrm{b}', 4'\mathrm{e}'\} \to '\mathrm{d}' \tag{12}$$

We will assume that the string 'd' rule is *systemically necessary*. This means that if 'd' is delivered to the controller below a minimum number, the whole supersystem will be dissolved (not just a system). This could be expressed as the controller having a special necessary rule that we could write as:

$$w_{n3S} : 'd' \to '' \tag{13}$$

If the controller is not provided with the number of input strings required for the minimum flow declared for w_{n3S} then the whole system would be dissolved!

A productive structure describes a set of rules so interconnected that a set of non-null strings is produced. To completely define a productive structure for the above example we set rules for the production of 'a' and 'b' as:

$$\begin{aligned} &w_a : * \to '\mathrm{a}' \\ &w_b : 3'\mathrm{e}' \to '\mathrm{b}' \end{aligned} \tag{14}$$

The vector net flow of such structure $\boldsymbol{y}(k)$ are the numbers of strings that the structure outputs. The outputs can go to the controller, to necessary rules or to other productive rules.

Assume that the nominal values of output flows $\boldsymbol{y}(k)$ are known. We will say that the structure is complete if actual steady-state operation of all rules is possible at nominal flow. Alternatively, we can say that no flow of some string is lacking for some other string to be at nominal flow.

In the example, we will assume a complete structure where all nominal flows are twice the minimal ones. The minimum flows or numbers of operation of the rules w_{n1}, w_{n2} will be 1 and 2. This means that one copy of 'e' and two copies 'f' are required for each system in each period. The overall system requires a minimum flow of 5 copies of 'd' in each period.

We will also assume that production of each type of string will be concentrated in one producer – no spread among several systems. Therefore, the supersystem will have five producing systems, corresponding to 'a', 'b', 'd', 'e' and 'f'. With these assumptions we can state the required nominal input and output flows in Table 1, where *NS* means the number of producing systems having an instance of the rule. The nominal flows are twice the minimum ones.

Note that if we would consider the controller as a system in the supersystem, there should exist in Table 1 a row with the rule w_{n3S}. For the moment, we are considering the controller as external to the supersystem.

The settings are such that a fully used or no-slack structure results: no string is produced in excess of what is required at nominal values.

Table 1. Determination of nominal input and output flows in example.

Rule	*NS*	*ynom*	*unom* ‘a’	*unom* ‘b’	*unom* ‘e’	*unom* ‘f’
w_d	1	10	20	30	40	
w_{n1}	5	2			10	
w_{n2}	5	4				20
w_a	1	20				
w_b	1	30			90	
w_f	1	20			40	
w_e	1	180				
Total outputs required:		10 (‘d’)	20	30	180	20

One must remember that a “no-slack” structure as this one requires “perfect” distribution to be complete, that is, strings must be delivered to systems in the exact fraction of their minimal or nominal requirements and not less, otherwise “lack” of input strings will ensue.

If all the producers are at nominal conditions and there are no disturbances, then ***S*** will be in a steady state, the nominal one. We consider now the problem of determining which assumptions are required in prices, taxes and their rules to have a steady state in numeraires, prices and taxes.

Table 2. Nominal sales of output and input flows.

System producig	must sell	must buy			
	ynom	*unom* ‘a’	*unom* ‘b’	*unom* ‘e’	*unom* ‘f’
‘a’	20			2	4
‘b’	30			2+90	4
‘d’	10	20	30	2+40	4
‘e’	180			2	4
‘f’	20			2+40	4

For this we assume as above that numeraire values are given as a function $N(i, k)$ of systems and time. We will assume that desirable constant positive nominal values for numeraires have been defined and that these values are the actual ones: $N(i, k) = N_{nom}$. If the nominal values are desirably constant, N_{nom} must be a steady-state or fixed-point:

$$N(i,k+1) = N(i,k) = N_{nom} \tag{15}$$

These makes sense as the productive part of the system is assumed at a steady state.

Now, let us consider the sum of all numeraire values:

$$N_0(k) = \sum_{i=1}^{I} N(i,k) \tag{16}$$

If the value of N_0 is to be maintained constant then the numeraire sourced by the controller as payments $P(k)$ must equal the numeraire drawn in taxes $T(k)$. In fact, we have the following updating equation for N_0:

$$N_0(k) = N_0(k-1) + P(k) + T(k) \tag{17}$$

This can be attained by the taxing rule presented in the sequel because it puts all systems in a numeraire steady state. It is clear that for any system, the variation of numeraire can only come from three flows: payments received by *i*, $p(i, k)$ with positive sign, purchases paid by *i*, $r(i, k)$, and taxes paid by *i*, $t(i, k)$, both with negative sign. The definitions for these are, taking *C* as the index of the controller:

$$\begin{aligned}
p(i,k) &= \left[\sum_{j=1}^{I} n(i,j,k) \frac{1+\operatorname{sgn}\big(n(i,j,k)\big)}{2} + n(C,i,k) \right]_{|n>0} \\
g(i,k) &= \sum_{i=1}^{I} n(i,j,k) \frac{1-\operatorname{sgn}\big(n(i,j,k)\big)}{2} \\
t(i,k) &= n(i,C,k)
\end{aligned} \tag{18}$$

The sign expressions are used to distinguish payments received from purchases paid.

It is clear that

$$N(i,k) - N(i,k-1) = p(i,k) + g(i,k) + t(i,k) \tag{19}$$

Therefore, if variation is to be avoided along a period one must set taxes to the symmetrical of the net sum of received payments and purchases:

$$\begin{aligned}
0 &= p(i,k) + g(i,k) + t(i,k) \\
t(i,k) &= -\big(p(i,k) + g(i,k)\big)
\end{aligned} \tag{20}$$

This is necessary in order to keep $N(k) = N_{nom}$. Yet, a previous assumption for keeping the flows at $Y(k) = Y_{nom}$ requires that systems can buy (in steady-state) all the strings they need. We enquire now if this is possible and it can be encoded as a set of rules. In Table 2, we list for each producer the nominal flows that must be sold and bought.

Systems can buy (in steady-state) all the strings they need if prices are such that each producer receives as payments an amount greater or equal than it pays in purchases to other producers. If the two amounts are equal, no taxes will be drawn, so we consider this case.

Let *rd*, *ra*, *rb*, *rf*, *re* be the prices of the strings to be determined. We get the following matrix equality:

$$\begin{bmatrix} 20ra \\ 30rb \\ 10rd \\ 180re \\ 20rf \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 92 & 4 \\ 20 & 30 & 0 & 42 & 4 \\ 0 & 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 42 & 4 \end{bmatrix} \begin{bmatrix} ra \\ rb \\ rd \\ re \\ rf \end{bmatrix} \tag{21}$$

We can spot that the conditions for the *re* and *rf* values are inconsistent. Therefore, we lift the restriction on zero taxes and assume that the unit cost and the unit price for the ‘e’ strings are equal to 1. We fix the other prices as the sum of the unit costs of the required input strings for the necessary and producing rules. Costs are functions of a constant term and a term proportional to output. They can all be expressed in rational multiples of the price of ‘e’. For the nominal flows supposed, the multiplier constants are given in (22). These are also the nominal prices for *re* = 1.

$$Y = Y_{nom} \rightarrow \begin{bmatrix} ra \\ rb \\ rd \\ re \\ rf \end{bmatrix} = \begin{bmatrix} 0.625 \\ 3.417 \\ 16.75 \\ 1 \\ 2.625 \end{bmatrix} re \qquad (22)$$

With the nominal prices set as in (22), only the producer of the 'e' string will pay taxes and the system will be in a steady state both in flows of strings as in flows of numeraire.

5.2 The effect of disturbances

We consider disturbances as changes in nominal values of prices or flows, in the state of affairs presented. We will assume that the goal of each coordinated system will be to keep or increase the numeraire it owns as this maximizes its expectance of survival.

Let the flows be constant and assume that a producer considers deviating the price from the nominal value at period *k*. Because the expectance of survival is assumed to be a measure of numeraire owned and the controller forces a constant nominal numeraire there is nothing to be gained from increasing or decreasing prices for any producer. Eventually, a deviation of price could imply loss of numeraire.

Given that the tax rule makes disturbances in prices a loosing decision, we consider now disturbances in flows.

We consider that a flow disturbance $\Delta y(i)$ can be either negative or positive with the possibility that the producer can or cannot do something about it.

We consider first that the producer could not do something about. If $\Delta y(i)$ is negative, this means that $y(i)$ and the flows dependent on $y(i)$ will decrease. In this case, the task of the controller is to avoid the numeraire rules producing undue damage. It can be seen in the example that the producer of 'a' will lose numeraire or disappear if the flow of 'a' decreases. This can be avoided if producers are allowed to increase the prices so to earn the same quantity of numeraire.

If the disturbance is positive, then prices can lower and other flows can increase, adjusting prices down.

Now, we consider the situation where the producer could prevent the lowering of the flow without stress – the case of the "lazy" producer. The tax rule must penalize the producer, decrease their nominal value of numeraire and this requires a tax greater in modulus than the difference between payments received and purchases paid. Some provision for allowing lazy producers to get back to nominal numeraire must also be designed.

6. CONCLUSIONS AND PERSPECTIVES

6.1 A wider view of the presented subjects

The goal of this paper is to set foundations for a theoretical framework addressing systems partially coordinated by a controller using a numeraire. The long-term purpose is to enlarge control theory in order to include coordination of systems or agents induced and mediated by the use of numeraires.

We began by presenting the concept of control of a system variable $x(t)$ as the constraining of its behavior to a proper subset of its behavioral space $X(t)$.

The concept for coordination taken here is to understand coordination as causation. This reduces the behavioral space of coordinated variables, a useful property in building a unified theoretical framework for coordination and control. It may be argued that for two causally related systems to be coordinated, one may require that something "desirable" comes out from the causal relation. This is a legitimate perspective of course, but it overlooks the fact that systems may exhibit coordinated behavior with undesirable results, for example, two armies at war or a group of technicians provoking a disaster by following – or not following – rules. If we are going to develop a framework that allows us to understand "coordination failures" or "coordination disasters", we need a concept of coordination that allows for both types of outcomes.

Surely, the most usual and pressing question about coordination is if we have a group of people, machines, or people and machines, that have *the potential capacity* for coordinating to attain a goal, how does one induces or organizes such coordination? To answer this question, one must consider another one: how does it happen "naturally"?

There are many possible answers for these questions. Wang and Saridis [7] illustrate an approach, based on a theory for intelligent machines where coordination results from a specific design of the coordinating systems.

For this paper, the work of Tumer and Wolpert [8], [9] was inspirational. They define systems, "where each agent aims to optimize its own performance, but where there is a well-defined set of system-level performance criteria", as *collectives*. In [1] Tumer and Knudson state:

"*Given some system level objective function [...], we aim to derive an objective function for the agents in such a way that when they achieve their own objectives, the system objective is also achieved.*"

Although Tumer and Wolpert speak of pay-off functions and give examples of the application of the their framework to economic exchanges, it is not clear that they understand as a quantity akin to money the values of the individual utility or objective function that each agent tries to maximize.

Szuba [10] observed that in simulations of the generation of collective intelligent systems a quantity exhibiting properties of money appeared recurrently. This was an important step, yet not enough to come to the formal idea, presented in this paper, of systems that coordinate their members through a quantity whose value may be interpreted as a criterion, that can be exchanged and which application is managed by a central controller.

This last step came out from taking acquaintance with a less known school of economic thought called chartalism, neo-chartalism or modern money theory. It was originated by Knapp [11] who understood as money whatever a state accepts for the payment of taxes. The concept was subsequently developed by Lerner [12] that view money, its creation and destruction, as a control variable in an economy, driven by the government. Actual authors in this school are Mosler [13] and Wray [14].

This acquaintance convinced this author that it is possible to abstract what has been called economic phenomena involving money as systems coordinating through numeraires. The potential importance of this abstraction goes well beyond economic phenomena, as the examples of complex technical systems given by Tumer and Wolpert show.

One may wonder if exploring the concept of systems coordinating through numeraires in terms of application to societies will do more than duplicating common knowledge in economics. In the opinion of this author, this is improbable. Consider the definition of money given by prestigeful mainstream economist Mankiw in [15]:

"*Money is the set of assets in the economy that people regularly use to buy goods and services from each other*."

If this concept of money is used to study systemic coordination in economic phenomena, it is difficult to see how it can lead to the results that one must obtain if one interprets money as a numeraire – a device to induce coordination.

The exploration of the idea of supersystems coordinated through the use of numeraires requires that the workings of what is *actually* the object of coordination, the behavior of systems in the supersystem, be represented and simulated. The choice made here was to suppose that each system is a string rewriting system with several production rules. There were two reasons for this. First, strings can be easily interpreted to mean whatever one intends. Second, string rewriting systems can be made computation universal, [16], [17], which means that, in principle, by writing production rules, one can model and simulate whichever system coordinated through numeraire one may be interested.

The abstract 'toy' example system presented in the previous section illustrated a classical approach of analysis in control. We considered a fundamental control strategy or tax rule that forced the numeraires and flows of all producers to be at a nominal steady state and assessed the effects of disturbances, assuming defined prices for the nominal flows.

This worked example may well give the reader a flavor of representing producing systems through string re-write rules and the workings of systems coordinated through numeraires. It suggests also several research lines.

5.3 Lines of development and research

In terms of producing structures, recursive rules and recursive structures, i.e., rules that use an output as some input, are to be formalized. Still, in this area, the relation of the representing method with deduction, computation and linear logic is to be understood. It would also be useful to determine in a systematic fashion the steady-state flows implied by the instances of the rules that are present in the system. The same could be said of analyzing and establishing nominal steady-state prices corresponding to nominal flows.

The main property of the tax rule considered is that it keeps the numeraire of agents in a constant steady state. Curiously, the rule is consistent with not penalizing producers when disturbances are not their fault, via adjustment of prices. However, an extra term in the rule must be considered to penalize producers that decrease flow by "laziness". We note that for this to work the controller must know when a disturbance is a fault of the producer or not. In a perspective of practical application, this is a very strong assumption.

Tax rules that increase or decrease the numeraire of producers appear as the field to explore next. Results in this field are necessary for actual practical application and theoretical development of the paradigm sketched here. Both must consider many real-world complexities that are inescapable:

– Dynamics of population and organizations. Note that while the example presented here is of an organization, these are made of individual agents that are created, persist and cease in time, or, if living beings, are born, live and die.

– Parallelization of production among organizations with cooperative and competitive behavior.

– Evolutive and learning organizations.

If the results of this research are to help in the analysis of economic phenomena, we must able to model and understand in the paradigm:

– The credit process, credit organizations and the financial system.

– The role of states as controllers. States have a duty of performing well as controllers so that the well being of people in the system is at least satisfactory.

ACKNOWLEDGMENT

Paulo Garrido was financially supported by FCT, Fundação Portuguesa para a Ciência e a Tecnologia, through the Program PEst, Strategic Program of the Algoritmi Research Center, Project FCOMP-01-0124-FEDER-022674.

REFERENCES

[1] Tumer, K. and Knudson, M.: Aligning agent objectives for learning and coordination in multiagent systems. *PerAdA Magazine*, IEEE, Computer Society, Oct 2010.

[2] Putnik, G., Škuljb, G., Vrabičb, R., Varela, L., Butalab, P. Simulation study of large production network robustness in uncertain environment. CIRP Annals - Manufacturing Technology (2015), http://dx.doi.org/10.1016/j.cirp.2015.04.118

[3] Ashby, W. R.: *An introduction to cybernetics*, Chapman and Hall, London, 1956.

[4] Ashby, W.R.: Requisite variety and its implications for the control of complex systems, *Cybernetica* 1:2, p. 83-99, 1958. (available at http://pcp.vub.ac.be/Books/AshbyReqVar.pdf.

[5] republished on the web by F. Heylighen—Principia Cybernetica Project),

[6] Saridis, G. N.: *Entropy and control engineering*, World Scientific, Singapore, 2001.

[7] Olson, G. M., Malone, T. W., and Smith, J. B. (eds.): *Coordination theory and collaboration technology*, Lawrence Erlbaum Associates, 2001.

[8] Wang, F.-Y. and Saridis, G. N.: A coordination theory for intelligent machines, Automatica, Vol. 26, No. 5, pp. 833-844, 1990.

[9] Tumer, K. and Wolpert, D. (eds): *Collectives and the design of complex systems*, Springer-Verlag, New York, 2004.

[10] Tumer, K. and Wolpert, D.: The science of collectives, in: Tumer, K. and Wolpert, D. (eds): *Collectives and the design of complex systems*, pp. 1-42, Springer-Verlag, New York, 2004.

[11] Szuba, T.: *Computational collective intelligence*, Wiley-Interscience, 2001.

[12] Knapp, G. F.: *The state theory of money* (English translation), Macmillan, London, 1924.

[13] Lerner, A. P.: *The economics of control*, Macmillan, London, 1946

[14] Mosler, W.: *Soft currency economics II*, Valance, (distr. by Amazon), 2013.

[15] Wray, L. R.: *Modern money theory*, Palgrave Macmillan, 2012.

[16] Mankiw, G.: *Principles of macroeconomics* (6th edition), South-Western Cengage Learning, 2011.

[17] Post, E.: Formal reductions of the general combinatorial decision problem, *American Journal of Mathematics*, Vol. 62, No. 2, pp. 197-215, 1943.

[18] Wikipedia: Semi-Thue system. https://en.wikipedia.org/wiki/Semi-Thue_system

КООРДИНАЦИЈА СИСТЕМА ПРЕКО РАЗМЕНЉИВИХ ВРЕДНОСТИ (NUMERAIRES)

П. Гарридо

У раду се предлаже парадигма координације система помоћу контролера који има довољно “моћи” да побуди координацију међу системима у односу на неки циљ.

Моћ контролера, који се у раду разматра, односи се на расформирање организације или елиминисање система, условне за вредности скаларног критеријума разменљивости који контролер примењује на систем. Критеријум се зове разменљиве вредности (numeraire), јер је парадигма преузета из економских система који се базирају на новцу.

У раду се даје опис метода компјутерског приказивања производних система који се заснивају на правилима за преписивање низа. Затим се преко разменљивих вредности (numeraire) проучава проблем регулације на једном апстрактном примеру координисаног система. Прво се одређује и активира номинална операциона тачка, а потом се проучавају ефекти поремећаја и шта контролер може да уради да их минимизује.

Иако је пример једноставан, он је поучан и омогућава истраживање у неколико праваца.