2019 18th European Control Conference (ECC)
Napoli, Italy, June 25-28, 2019

Control-oriented modelling of proof-of-work blockchains

Alberto Leva, Silvia Strada, Mara Tanelli

Abstract— **Blockchain (BC) technology is a rather new conception of a mixed hardware and software platform to achieve distributed consensus among peers. Its diffusion is related to cryptocurrency, the most widespread of which is Bitcoin. The protocol on which BCs operate sees the interaction between users, interested in performing their transactions, and miners, who certify the trust behind the transactions by putting some form of effort that allows acknowledging their trustfulness, obtaining Bitcoins in reward for their work. In the so-called proof-of-work implementation of the BC, such effort is the computational power needed to find a specific string of bits called *nonce*. The resulting game-theoretic setting has subtle dynamics, and its functioning could be strongly improved using closed-loop control. This work is an attempt to define a control-oriented description of the agent-based BC dynamics and offer a redesign of the *difficulty* control system that regulates the amount of work needed to add a new trusted block to the BC. This control loop directly relates to the energy consumption of the overall system, which is one of the major drivers that will determine the future sustainability of the BC paradigm.**

I. Introduction

The blockchain (BC) technology itself is nowadays considered a real and important innovation, much more than just the software mechanism behind cryptocurrencies, see for example [12]. As a matter of fact, there are many other areas of society where the blockchain rationale could have an incredible potential, for example in the industry, in the public and legal sector and in general in the internet of things. Basically, the BC is an electronic register of digital records, events or transactions that are managed by the participants to a distributed network, see *e.g.*, [3]. Thus, the BC is made of both a data structure and of the associated management system. Originally, the BC was proposed by [8] as a method to securely time-stamp a document and link it to the next one inline to ensure the chronological order of the database. The intrinsic security of the linking procedure is guaranteed by the so called hash function, that maps any record to a fixed length alphanumeric string. This sequence of records in chronological order is spread out to all the users of the system, so mitigating failure risks. This idea of distributing a chain of securely connected entities is now widely known as decentralized consensus ledger technology. After its initial stages, BC was implemented to keep track of money transactions in a digital, non-intermediated setting by a person known under the pseudonymous of Satoshi Nakamoto, the creator of Bitcoin, [19], the first BC based cryptocurrency. Specifically, all the entities participating in the BC network are allowed to verify transactions and some of them, the miners, compete to solve a mathematical puzzle in order to create a block for a given transactions set. The process of generating a new block is called "mining". In order to mine a block, the miner needs to show his work and the related procedure is called proof of work. Proof of work means that a miner has to carry out a really hard task, *i.e.*, finding the mapping to the alphanumeric string, that is easy for somebody to verify once the miner has done it. Think of proof of work like a hard labor task, for which *hashing power* is needed, but once the miner has completed the job, it is easy for all involved to verify that a very precise task was fulfilled. Mathematically speaking, the proof of work is a random process with low probability, so that a lot of trial and error is required, on average, before a valid hash value is generated. When a block has been mined, miners share it and, if they get consensus of other miners, the block is added to the BC and its hash value written in the header of the next block. Each time a block is mined and attached to thee BC, successful miners are rewarded for their work with coins of digital currency. The amount of the reward varies for different digital currencies, see *e.g.*, [24]. BCs are opening a wealth of possibilities, but at the same time entail a tough energy challenge: should Bitcoin – or proof-of-work BCs at large – become mainstream, the sustainability itself of the electrical system could be endangered [4], at least with the present technology in place. Adopting a dynamic modelling and control viewpoint numerous opportunities appear for both mitigating the problem and improving the BC implementation paradigm in general. This paper aims at taking the challenge above with a system- and control-theoretical attitude, which to date is far from mainstream in the design of BCs, with the purpose of starting a long-term research. In this first work propose a possible BC modelling framework along those principles, and some preliminary tests to support our position. More specifically, the paper is organized as follows: Section II presents a brief review of the literaure on blockchain models, while Section III illustrates the proposed multi-physical model of the BC proposed in this work, and it formulates the difficulty control problem, offering a genuine system-theoretic solution to be compared to the most currently widespread one. The simulation examples presented in Section V prove the potential of the closed-loop approach in terms of reduction of energy consumption.

II. Brief literature review

Blockchain technologies have raised considerable attention in the scientific community since their diffusion, started in 2009, motivated by the creation of Bitcoin, [19]. Modelling such a system has been attempted in different communities, mainly Computer Science, interested in the implementation-

A. Leva, S. Strada and M. Tanelli are with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo Da Vinci, 32 – 20133 Milano, Italy; {alberto.leva,silvia.strada,mara.tanelli}@polimi.it

978-3-907144-00-8 ©2019 EUCA

related description of its functioning, see *e.g.*,O [16], and Economics and Finance, looking more at the economic ecosystem enabled by the technological layer and its financial sustainability mechanisms, see *e.g.*, [11].

In this work, we are particularly interested in analysing the modelling efforts that have been made to describe the blockchain dynamic evolution over time. In this respect, not many systematic results are available yet. The most relevant contributions trying to formally describe the BC functioning principles in a manner that is close to system-theoretic methods are the following. In [6], the authors study the effect of communication delay on the evolution of the BC over time, within a Markovian modelling framework, under the assumption of the so-called selfish-mine strategy adopted by the miners. By doing so, they find that both dishonest and honest miners are damaged by the presence of the selfish set, and that the dishonest behaviour can be detected monitoring the rate of production of orphan blocks (*i.e.*, blocks that form an abandoned fork end of the BC; such a fork occurs when disagreement exists among miners about which is the trusted new block to add to the precedent history of the BC).

A game-theoretic approach is used in [15], to study the large variance in rewards across miners that commonly exists in Bitcoin BC and the resulting formation of mining pools, which miners unite to form in order to ensure themselves a steady income stream by sharing the rewards among the pool members. More specifically, cooperative game theory is employed to study which pools agents may wish to join, and how pool members are likely to share the monetary rewards. The authors show, under this modelling framework, that achieving a stable distribution of rewards is unlikely, and that, should Bitcoin become widespread, there a larger share of miners would switch among pools over time to improve their reward, resulting in a larger overhead for the BC itself.

Further, [11] describes the problem difficulty control as it is commonly addressed in the Bitocoin proof-of-work BC, and studies the reaching of an equilibrium condition between miners and users that is based on an optimal trade-off between the fees that users attribute to each transaction and the mining cost. Finally, [10] provides a genuine control-theoretic approach to study the problem of difficulty regulation in the mining process using as feedback variable the average time needed to mine a block over a certain time period, even though based on a very simplistic, and mostly static model of the BC evolution. With respect to this work, we propose a more detailed dynamical model of the BC dynamics that captures all the interactions among the agents involved in the process, and which identifies the inputs and outputs variables that define the interfaces between the different systems components, introducing also the crucial aspect of the energetic costs associated with the block mining activity. Not surprisingly, in fact, as far as practical sustainability and evolution perspectives of the proof-of-work BC system are concerned, the main issue is that of its energy demands. It is recognised, in fact, see *e.g.*, [4], that if Bitcoin would become mainstream, the amount of electrical energy needed to feed the resulting system would be overwhelmingly large. Such energy consumption comes from the computational power needed to mine the blocks. Previous work has demonstrated that the use of closed-loop control systems based on appropriate dynamic descriptions of the underlying systems can indeed provide energy savings and a better use of resources, [14], [13], [20], [22], [21]. Therefore, a first effort should, in our view, be devoted to explicitly model the energy consumption mechanisms that are involved in the blockchain functioning. Based on these models, it would then be possible to prove how closed-loop control may help in ensuring a lower and, most importantly, an *a-priori* defined energy consumption level, and in having degrees of freedom to trade-off the speed at which the blocks are mined with energy-related constraints. In this work, we take a first step in this direction, defining a cyber-physical (CP), agent-based description of the BC dynamics, and introducing simple, yet effective control loops that allow us to show the potential of applying control-theoretic principles to this very interesting and challenging system.

III. First-principle multi-physics BC modelling

This section builds upon two existing modelling works [10], [11], and [18], to which the reader is referred for more details on the BC operation. With respect to literature models, extensions are here proposed for both the BC behaviour, and the control that it should be endowed with.

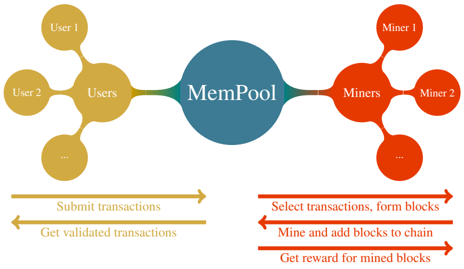
User 1
User 2
Users
...
MemPool
Miners
Miner 1
Miner 2
...
Submit transactions
Get validated transactions
Select transactions, form blocks
Mine and add blocks to chain
Get reward for mined blocks

Fig. 1. High-level scheme of a BC MemPool.

A BC can be viewed as a set of *mining pools*, or *MemPools*. In this paper, with very small – if any – generality loss, we consider one pool. The operation of a BC MemPool is illustrated, at a very abstract level, in Figure 1. Users submit *transactions* to be *validated* so as to guarantee the truthfulness of their content. A transaction has a *size*, and the user offers a *reward* for the service of validating it. Miners select transitions from the pool and group them to form a *block*. To this end they use heuristics, in an attempt to earn the most, according to their memory and computational capacity (*hashing power*). A miner attempts to *mine* a block, *i.e.*, to complete the block with an alphanumeric string called the *nonce*, which has to enjoy some common properties with the hash. The way these properties are set forth dictates the mining *difficulty*. The operation of the MemPool is organised in *periods*. A new period starts when a fixed number N of blocks are mined. For example, for Bitcoin BC the period is $N = 2016$, but for example for another blockchian implementation called the Bismuth BC $N = 1$. At the beginning of a period, the difficulty is defined (*via* a consensus mechanism inessential to be discussed herein) and then kept constant along the whole period.

For our purposes, the user-pool-miner chain is described by the three automata in Figures 2 through 4. The automaton in Figure 2 represents a user, who adds transactions to the chain with a certain time distribution: at time t_{next} the next transaction is submitted. We assume an exponential distribution of the transactions' deposit, for consistency with the typical distribution of job arrival times in queueing networks [2], but this assumption could be worth of further investigation.

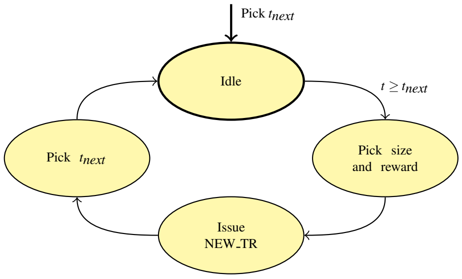

Fig. 2. User model automaton.

The automaton in Figure 3 describes the pool (actually a distributed database, but this is not relevant for the behavioural description that is the purpose of our control-oriented model). The automaton operation should be self-explanatory. The green node **Update Difficulty**, where the difficulty of the next period is determined, implies some feedback control. With the present state of the art, the typical objective is to maintain the average block mining time to an *a-priori* fixed value (as an example, for Bitcoin such mining time is 10 minutes), so as to smooth out possible takeover attempts performed by abruptly bringing in a large hashing power.

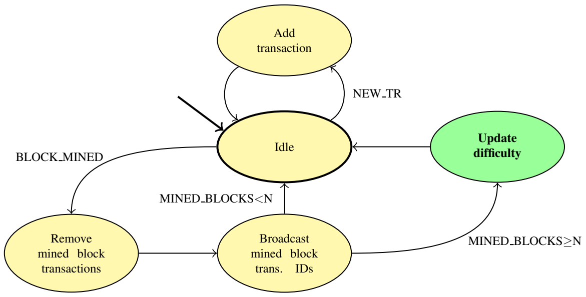

Fig. 3. MemPool model automaton.

The automaton in Figure 4 represents the miner, and its operation should be self-apparent as well. The orange node **Select Transition** contains the heuristics used to select the transitions to be included in the forming block. The green node **Hash** should, in our opinion, include some feedback control to manage computational resources. To the best of our knowledge, this matter is not addressed in the BC literature and it is just left to the operating systems and/or the "load balancer" aboard the miner's machine(s), *i.e.*, to software components that in general are not designed having the peculiarities of BC-induced loads in mind.

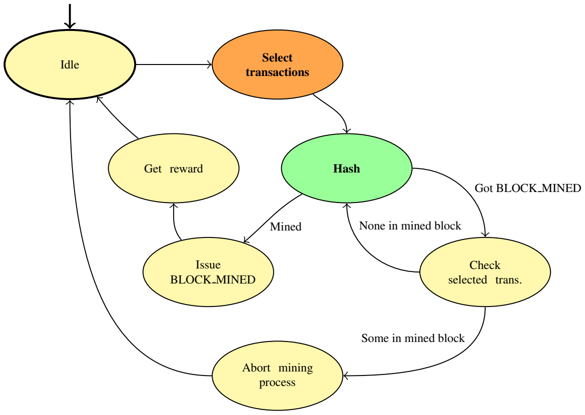

Fig. 4. Miner model automaton.

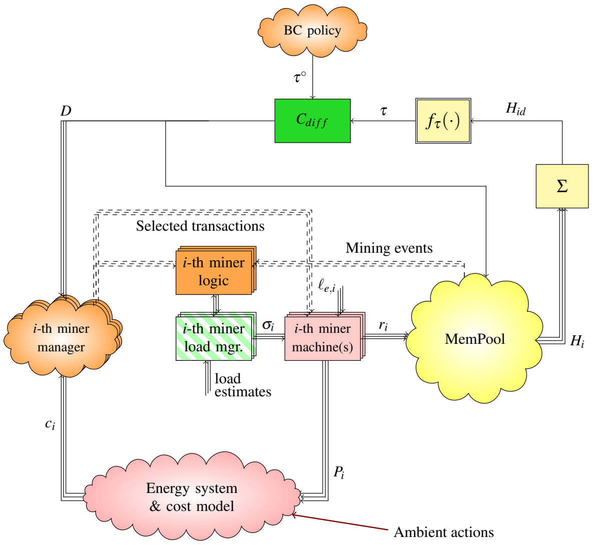

Fig. 5. Behavioural model of a BC as a CP system (state of the art).

Overall, the behaviour of a BC is represented by the Cyber-Physical (CP) system depicted in Figure 5, where solid and dashed lines correspond to numeric and lexical signals, respectively. As it is common in IT control systems, the CP and the Controller-Plant partitions do not coincide. The controlled system (the plant) is composed by the miners' machines and the energy system, whatever the latter means, and this is physical. The plant has however a cyber part as well, *i.e.*, the immaterial dynamics of the MemPool. We can thus talk about a P-plant and a C-plant, denoted in Figure 5 by the magenta and the yellow colour, respectively. The P-plant inputs are the resources allotted to mining, such as CPU/GPU shares or number of active ASICs in the case the miner uses specialised hardware [23], here collectively denoted by σ_i, and where applicable the exogenous loads from other processes, the operating system and so forth, plus any disturbance affecting the relationship between allotted resources and accomplished computational work; all such disturbances are here indicated by $\ell_{e,i}$. The P-plant outputs are the machine computation rates r_i, their power consumptions P_i, and the computation costs c_i. This is a peculiarity of our approach, as literature models do not address energy-related facts and take the costs as exogenous inputs. The relevance of mining on grid management, see [4], makes such approach highly questionable.

The C-plant inputs are the computation rates r_i and the difficulty D, that result in the miners' hashing rates H_i. This is another very important characteristics of our approach, as literature models normally take the hashing rates (or powers) as exogenous inputs, while no controller can prescribe them if not through the dynamics of the computing system. Many computer- related control papers do not address this dynamics, but for demanding problems doing so is very questionable as well. One does not prescribe computation or service times, but rather requires the operating system to allot certain resource shares, which in turn result in certain computational performance. Hence, service times – thus hashing powers – are a consequence, not an input. Thus, a probabilistic setting is used, which depends on the miner working condition. In detail, we assume that the time τ_m to mine a block is governed by a Gumbel distribution, [7], [1]. This distribution turns out to be useful to more realistically model the distribution of the maximum value within sample data, if their underlying distribution is of the normal or exponential type, the latter being the widely accepted model for mining times in BCs. Gumbel has shown that the maximum value in a sample of a random variable following an exponential distribution approaches the Gumbel distribution as the sample size increases, [7]. The Gumbel cumulative distribution function is

$$c(\tau_m) = e^{-e^{-\frac{\tau_m - \alpha}{\beta}}}, \tag{1}$$

and the distribution mean μ_{τ_m} and standard deviation σ_{τ_m} are given respectively by

$$\mu_{\tau_m} = \alpha + \beta\gamma, \quad \sigma_{\tau_m} = \frac{\pi}{\sqrt{6}}\beta, \tag{2}$$

where γ is the Euler-Mascheroni constant

$$\gamma = \int_1^\infty \left(\frac{1}{floor(x)} - \frac{1}{x} \right) dx = 0.57721... \tag{3}$$

To make τ_m dependent on the computation rates r_i and on the difficulty D, we let μ_{τ_m} and σ_{τ_m} depend on D and on the total processing rate of all the miners, as follows. First we compute the ratio R of the two aforementioned quantities, *i.e.*,

$$R = \frac{D}{\sum_i r_i}, \tag{4}$$

then we scale the distribution mean μ_{τ_m} and standard deviation σ_{τ_m} as

$$\mu_{\tau_m} = \frac{\overline{\mu}_{\tau_m}}{\overline{R}} R, \quad \mu_{\sigma_m} = \frac{\overline{\sigma}_{\tau_m}}{\overline{R}} R, \tag{5}$$

where overlined letters denotes values at a BC operating condition taken as reference, and finally we use μ_{τ_m} and σ_{τ_m} to determine α and β as per (2). The *rationale* is that speeding up a machine (or group of machines) with respect to the work to do, makes the completion time distribution both smaller in average and less scattered around the average itself. We deem this scaling technique adequate for the present level of the research, but the matter surely needs further investigation in the future.

Coming back to Figure 5, the hashing powers H_i are added to give the *maximum available* total hashing power H_{id}—maximum available because more than one miner can select the same transaction, and when a mining is successful the competing ones' power is lost. The overall average time τ for mining a block is therefore

$$\tau = f_\tau(H_{id}) = \frac{\delta_P}{H_{id}}, \tag{6}$$

where $\delta_P \geq 1$ is a variable gain – or equivalently, a multiplicative disturbance – interpreted as the pool (time-varying) keenness to wasting power due to overlapping minings. Random disturbances like δ_P are extremely hard to predict, so that previous studies on application progress control (a sibling problem to maintaining a mining pace) suggest to model them as just bounded, with no spectral assumption [13].

The control part of the system is composed of the BC-wide policy to determine the desired mining time τ°, of the miner managers' heuristics to select transactions, and of the miners' logic to react to events from the pool. The heuristic/logic part is distinguished in Figure 5 by the orange colour. There is also some modulating control. The dark green block C_{diff} is the feedback law to determine D, and operates at the time scale of periods. Most frequently a very simple nonlinear law is adopted, like

$$D(k) = D(k-1)\frac{\tau^\circ}{\tau(k)} \tag{7}$$

where k counts the periods. Moreover, as anticipated, the miner logic acts on computations through local resource managers or "load balancers". This non-strictly-feedback nature is the reason why the blocks in Figure 5 are striped, while the pale (not dark) green colour indicates that they operate at a significantly faster rate than the periods, most frequently as part of the local operating systems.

IV. Proposed control-based improvements

In Section III, we showed that a BC can be seen as a multi-physics CP system with multi-rate control. Adopting an object-oriented modelling paradigm [5], mixing equation- and algorithm-based components, is the natural way to turn this viewpoint into efficient simulation models. There is, however, another front on which control can provide contributions, and on this front we propose to modify the BC operation to reflect the CP model in Figure 6.

The first and simpler improvement we propose is to design the block C_{diff} as a feedback controller acting on the block mining time error $\tau^\circ - \tau$. This is not a complete

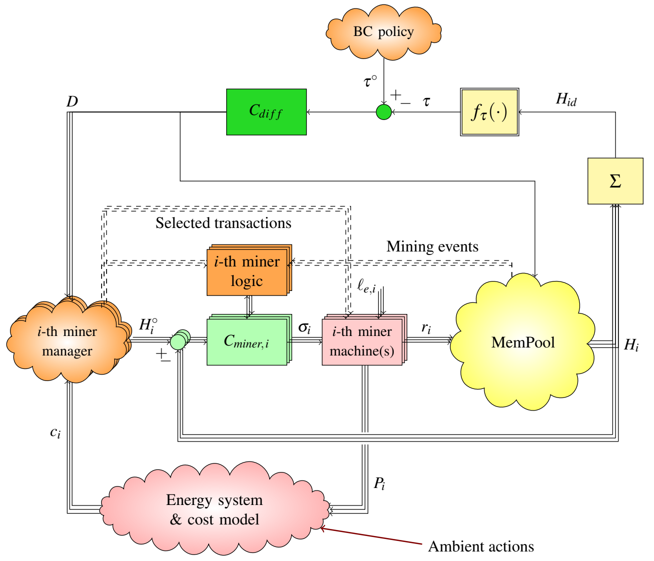

Fig. 6. Behavioural model of a BC MemPool as a CP system (proposed developments).

novelty [10], but we believe that the first-principle nature of our model allows more insight about the closed loop system operation. Second and most important, we propose to instrument the miners' client software, and close local loops (with controllers C_{miner}) in each of them, so as to maintain a certain hashing rate. This counteracts the fast variability of the pool, most likely easing the job of the difficulty controller. Technical solutions like [9] are available for instrumenting the clients, and their suitability to power-related controls is proven [17].

V. A Simulation Example

To show our control-oriented modelling approach at work, we now compare the standard control in Figure 5 with the law (7), denoted here by "ratio" control, against a solution adhering to Figure 6, where C_{diff} is a PI computing D based on the normalised error $e_n = (\tau^\circ - \tau)/\tau^\circ$, while the local miners' controllers are pure proportional ones that determine a computational power boost (with respect to a machine-wide baseline value C_0) based on the difference between the desired and actual mining time, computed – quite obviously – only when a mining is successful. The model was realised in Modelica, so as to be easily connected to physically heterogeneous one when the study will be widened to larger-scale energy systems, networks, and the like.

For space reasons we omit further details on the proposed controller tuning phase, which is not the main focus of this paper and will be discussed in future works. For the same reason we show just one test, where the BC pool model is subject to a negative variation of C_0 at 10000 time units, and to a decrement of the desired average mining time at 20000 time units.

Figure 7 shows the normalised error, which is comparable with the two controls, and the corresponding difficulty, normalised as well for readability in the [0,1] range. As can be seen, the proposed control structure – the merit resting in particular with the miner-level loops, for the sake of precision – yields its result with a consistently smaller difficulty, hence spent power, hence consumed energy and cost. This is further evidenced in Figure 8, that reports the normalised difficulty difference and its time integral. As can be seen, although numbers are not discussed as the focus here is on the enabling modelling approach, the saving is apparent.

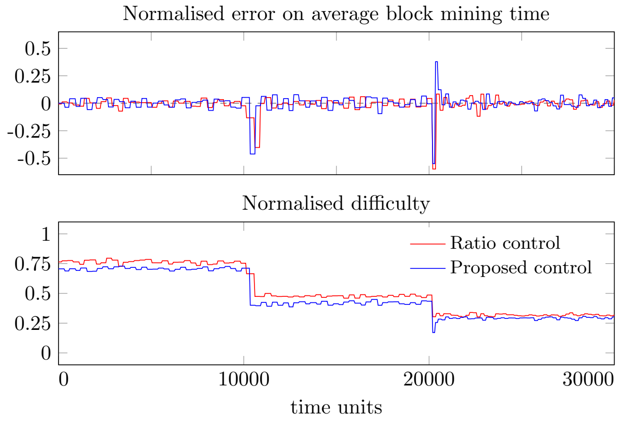

Fig. 7. Simulation example: time histories of the normalised errors (top) and control actions (bottom).

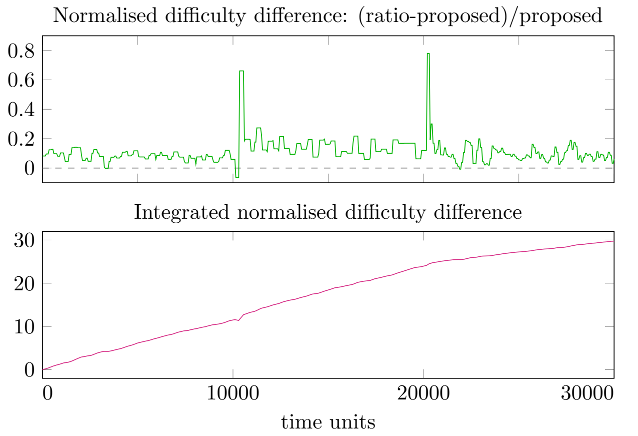

Fig. 8. Simulation example – control effort comparison.

VI. Acceptance Considerations

As discussed e.g. in [23], the heavy computational work of nowadays' BCs is fostering the growth of purpose-specific hardware. This is typically based on ASICs (Application-Specific Integrated Circuits), hence not on CPUs/GPUs managed by some operating system, and possibly shared by other tasks than mining. Moreover, in a proof-of-work BC, miners are competing with one another, their behaviour is totally selfish, and can be supposed to ignore any suggestion they do not foresee to entail more money earned.

In such a *scenario*, one may wonder whether any control aiming at an energy consumption reduction will be accepted or not. We give just a couple of examples why, although the matter strays from this paper. On the hardware side, mining ASICs are normally used full-throttle, as not doing so is seen by miners as not exploiting their investment. Since their load is thus almost constant and not shared by non-mining tasks, the need for miner-level controls may not be so evident. On the management side, big players may not like a control system to improve energy efficiency via an overall difficulty reduction, as the consequent reduction of hashing power requirements could allow smaller miners to (re-)join the game from which they were progressively expelled—thanks exactly to increasing energy needs, from the big ones' viewpoint.

Notwithstanding the expectable difficulties, however, we bear to state that the considerations above further support research on modelling and control. On the equipment front, sooner or later any hardware to stay on the market shall need to withstand enough stress to require control aboard. This happened to CPUs, there is no reason – with the due differences – for mining devices to not follow the rule somehow. On the management side, the game has in fact one more player than the miners. This is the manager of the grid, and is gaining importance. As the impact of mining (no matter with which hardware) becomes relevant, the manager may decide to not consider BC-induced load as any other load, and take action. This could be for example requiring a fee for mining, to participate in the gain as a reward for bearing the power burden, or differentiating the price on a time/location basis, or any other action that ends up contrasting the miners' desire/assumption of virtually unlimited energy availability. Clearly, such events would change the problem significantly.

In the authors' opinion, it is extremely difficult to figure out how the story will evolve. The only certainties are that control is going to play a relevant role, and also that, to take knowledgeable actions, all the involved entities need reliable models, capturing the relevant dynamic phenomena without undue complexity.

VII. Concluding remarks and outlook

This work considered the problem of modelling the BC dynamic evolution over time. Specifically, such modelling aimed at unveiling the cyber-physical nature of the system, and its interaction with its purely physical parts. This has been done within the realm of control theory, offering a means to define in which parts of the BC a closed-loop controller might provide beneficial effects. The preliminary results on difficulty regulation showed that promising improvements can be achieved, which can be directly related with energy savings, thus addressing the main issue in determining the future spread of BC technologies.

References

[1] J. Hansen and. Hissam and G.A. Moreno. Statistical-based WCET estimation and validation. In *Proc. 9th International Workshop on Worst-Case Execution Time Analysis*, volume 10, pages 1–11, Dagstuhl, Germany, 2009.

[2] D. Arcelli, V. Cortellessa, and A. Leva. A library of modeling components for adaptive queuing networks. In *Proc. 13th European Workshop on Performance Engineering*, pages 204–219, Chios, Greece, 2016.

[3] A. Berentsen and F. Schar. A short introduction to the world of cryptocurrencies. *Federal Reserve Bank of St. Louis Review, First Quarter 2018*, pages 1–16, 2015.

[4] P. Fairley. Feeding the blockchain beast – if bitcoin ever does go mainstream, the electricity needed to sustain it will be enormous. *IEEE Spectrum*, 54(10):36–59, 2017.

[5] P. Fritzson. *Principles of object-oriented modeling and simulation with Modelica 3.3: a cyber-physical approach*. John Wiley & Sons, Hoboken, NJ, USA, 2014.

[6] J. Göbel, H.P. Keeler, A.E. Krzesinski, and P.G. Taylor. Bitcoin blockchain dynamics: the selfish-mine strategy in the presence of propagation delay. *Performance Evaluation*, 104:23–41, 2016.

[7] E.J. Gumbel. *Statistical theory of extreme values and some practical applications. Applied Mathematics Series.*, volume 33 (1st ed.). 1954.

[8] S. Haber and W. S. Stornetta. How to time-stamp a digital document. *Journal of Cryptology*, 3:99–111, 1991.

[9] H. Hoffmann, J. Eastep, M.D. Santambrogio, J.E. Miller, and A. Agarwal. Application heartbeats: a generic interface for specifying program performance and goals in autonomous computing environments. In *Proc. 7th international conference on Autonomic computing*, pages 79–88, Washington, DC, USA, 2010.

[10] G. Hovland and Kučera J. Nonlinear feedback control and stability analysis of a proof-of-work blockchain. *Modeling, Identification and Control*, 38(4):157–168, 2017.

[11] E. Iyidogan. Economic model of blockchain based cryptocurrencies. Available online at `https://ssrn.com/abstract=3152803` or `http://dx.doi.org/10.2139/ssrn.3152803`, 2013. Imperial College Business School report.

[12] N. Lachance. Not just bitcoin: Why the blockchain is a seductive technology to many industries. *National Public Radio Online*, May 2016.

[13] A. Leva, S. Seva, and A.V. Papadopoulos. Progress rate control for computer applications. In *Proc. 17th European Control Conference*, pages 3173–3178, Limassol, Cyprus, 2018.

[14] A. Leva, F. Terraneo, I. Giacomello, and W. Fornaciari. Event-based power/performance-aware thermal management for high-density microprocessors. *IEEE Transactions on Control Systems Technology*, 26(2):535–550, 2018.

[15] Y. Lewenberg, Y. Bachrach, Y. Sompolinsky, A. Zohar, and J.S. Rosenschein. Bitcoin mining pools: a cooperative game theoretic analysis. In *Proc. 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 919–927, Istanbul, Turkey, 2015.

[16] I.C. Lin and T.C. Liao. A survey of blockchain security issues and challenges. *nternational Journal of Network Security*, 19(5):653–659, 2017.

[17] M. Maggio, H. Hoffmann, M.D. Santambrogio, A. Agarwal, and A. Leva. Power optimization in embedded systems via feedback control of resource allocation. *IEEE Transactions on Control Systems Technology*, 21(1):239–246, 2013.

[18] Raheel Ahmed Memon, Jian Ping Li, and Junaid Ahmed. Simulation model for blockchain systems using queuing theory. *Electronics*, 8(2):1–19, 2019.

[19] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, http://bitcoin.org/bitcoin.pdf, 2009.

[20] A. Sawada, S. Nakamura, H. Kataoka, T. Enokido, and M. Takizawa. Algorithms to energy-efficiently select a server for a general process in a scalable cluster. In *Proc. 31st International Conference on Advanced Information Networking and Applications Workshops*, pages 138–145, Taipei, Taiwan, 2017.

[21] Mara Tanelli, Danilo Ardagna, and Marco Lovera. On-and off-line model identification for power management of web service systems. In *Proceedings of the 47th IEEE Conference on Decision and Control, 2008. CDC 2008.*, pages 4497–4502, 2008.

[22] Mara Tanelli, Danilo Ardagna, and Marco Lovera. Identification of lpv state space models for autonomic web service systems. *IEEE Transactions on Control Systems Technology*, 19(1):93–103, 2011.

[23] M.B. Taylor. The evolution of bitcoin hardware. *Computer*, (9):58–66, 2017.

[24] F. Tschorsch and B. Scheuermann. Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys Tutorials*, 18(3):2084–2123, 2016.