Parallel Self Optimizing Control Framework for Digital Twin Enabled Smart Control Engineering

Jairo Viola and YangQuan Chen
Mechatronics Embedded Systems and Automation Lab
University of California, Merced
Merced, CA 95343, USA
{jviola,ychen53}@ucmerced.edu

2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI) | 978-1-6654-3337-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/DTPI52967.2021.9540078

***Abstract*—This paper presents a parallel Self Optimizing Control (SOC) framework that combines parallel intelligence, Digital Twin, and derivative-free optimization to enable smart capabilities in classic process control. The parallel SOC framework supports the interaction between the physical system and its Digital Twin via Simultaneous Perturbation Stochastic Approximation (SPSA) derivative-free optimization algorithm. The framework is tested using the Digital Twin of a thermoelectric heating system. Obtained results show that using parallel intelligence with Digital Twin and SPSA optimization method, the SOC can improve the system performance by introducing a developmental behavior on the classic process control system.**

***Index Terms*—Parallel Control, Self Optimizing Control, Simultaneous Perturbation Stochastic Approximation, SPSA, Digital Twin, Process Control.**

I. Introduction

The new manufacturing processes for Industry 4.0 require smart control capabilities that make systems aware of their current health status, performing developmental control actions that satisfy a desired performance cost function. In that context, novel technologies like Digital Twin (DT), Parallel Intelligence, and derivative-free optimization algorithms can support smartness in industrial process control through Self Optimizing Control (SOC) schemes.

The SOC is a control strategy designed initially for choosing the control variables for chemical plants control with tens or hundreds of these satisfying a cost function in terms of economic performance [1]. There are Different scopes of SOC like Extremum Seeking [2], Iterative Learning Control [3], or Run 2 Run control [4]. In these approaches, the controller parameters are adjusted according to the evolution of the economic cost function.

However, a SOC problem may take a long time before reaching an acceptable solution based on an economical cost function due to the nature of the optimization problem and the system dynamics, for example, thermal processes. In that case, parallel intelligence and control as a novel paradigm that looks for the integration of complex systems under the ACP approach (Analysis, Control, Parallel Execution) to improve the system performance can be integrated into the SOC to accelerate the system learning, and optimization [5]–[7].

In order to introduce parallel capabilities to SOC, a different optimization algorithm is required to handle the presence of multiple simultaneous executions to enhance the system performance and optimization speed. In that sense, the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm can be used [8], which is a derivative-free optimization technique that performs a stochastic approximation of the function gradient using only two measurements of the cost function. It has applications in controls for offline controllers tuning [9], [10] and feature extraction in machine learning [11], [12]. However, the most important property of SPSA is that it can be implemented for parallel execution [13], [14]. Thus, SPSA can be used as bridge between virtual and real systems to enhance SOC performance by simultaneous evaluation of multiple models of a system represented by instances of Digital Twins.

This paper presents a parallel Self Optimizing Control framework enabled by Digital Twins and the SPSA algorithm for the control of a stable closed-loop system based on an economic cost function. The framework uses the parallel implementation of the SPSA algorithm supported by a pool of Digital Twins of the real system to increase the optimization speed. Likewise, the SPSA handle the interaction between a physical system and a parallel Digital Twin, monitoring the closed-loop system behavior and updating the controller parameters according to an economic performance cost function. A Peltier thermoelectric system [15] is employed as a case study to evaluate the parallel SPSA framework. Two tests are performed for the system, one using only the real and virtual Digital Twin, and another with the support of the Digital Twin pool to leverage parallel capabilities of the SPSA.

The main contribution of this paper is proposing a parallel SOC framework using the SPSA algorithm, Parallel Intelligence and Digital Twin to perform smart control of a closed-loop system, enhancing its performance based on an economic cost function.

II. Parallel Self Optimizing Control Framework

The parallel SOC framework is shown in Fig. 1. As can be observed, a Parallel control architecture is employed, differentiating the real domain and virtual domain where the Digital Twin is located. In the real domain, a closed-loop system operates using a controller $C(\theta)$ which tuning parameters θ ensure the system stability. Likewise, In the virtual domain, the Parallel Digital Twin 1 replicates the configuration of the physical system using a multiphysics model of the process.

The goal of the parallel SOC framework is to continuously update the controller parameters θ based on an economic cost function, which considers the current performance of the closed-loop system. In this paper, the economic cost function employed is given by (1), where T_s is the system settling time, OV is the overshoot percentage, θ is a vector with the controller parameters, $W_{1,2,3}$ are the weights for the Overshoot, Settling time and the Integral Square Error index respectively.

$$\min_{\theta \in \mathbb{R}} J = W_1\ OV(\theta) + W_2\ T_s(\theta) + W_3 \int_0^t e(t,\theta)^2\,\mathrm{d}t. \quad (1)$$

The Simultaneous Perturbation Stochastic Approximation optimization algorithm is employed to find the optimal values of the controller tuning parameters θ. After each optimization step, executed in a different and higher timescale than the process closed-loop control, the result of the SPSA algorithm is updated to the Virtual and Real domains enabling a simultaneous interaction between domains. Considering that the SPSA algorithm can be executed in parallel, a pool of Digital Twins can be enabled in the virtual domain as shown in Fig. 1 to improve the convergence of the SOC. These Digital Twins act as slaves of the parallel system (Real system and Digital Twin in virtual system (DT#1), evaluating one of the multiple simultaneous perturbations required for the parallel SPSA optimization at each iteration, increasing the convergence speed of the algorithm. Thus, the SOC control using SPSA acts as an integrating rule for the parallel system.

A. Parallel SPSA Algorithm

SPSA is a stochastic optimization algorithm proposed by Spall [8], which considers the following optimization problem

$$\arg\min_{x \in \mathbb{R}^n}\ f(x), \quad (2)$$

that uses the recursive form of a general Stochastic Approximation algorithm:

$$x_{k+1} = x_k - a_k \bar{g}_k(x_k), \quad (3)$$

where x_k represents the estimate of x at the k-th iteration, where a_k is a sequence of positive scalar coefficients. So the approximation of the gradient at x_k is

$$\bar{g}_k(x_k) = \begin{bmatrix} \frac{f(x_k+c_k\Delta_k)-f(x_k-c_k\Delta_k)}{2c_k\Delta_{k1}} \\ \frac{f(x_k+c_k\Delta_k)-f(x_k-c_k\Delta_k)}{2c_k\Delta_{k2}} \\ \vdots \\ \frac{f(x_k+c_k\Delta_k)-f(x_k-c_k\Delta_k)}{2c_k\Delta_{kn}} \end{bmatrix}, \quad (4)$$

where n is the size of the input x, $\Delta_k = [\Delta_{k1}, \Delta_{k2}, \ldots, \Delta_{kn}]$ are the elements of the random perturbation vector Δ_k generated using a sub Bernoulli distribution, which are assumed to be independent and symmetrically distributed around zero, c_k is a positive scalar that change its value per each iteration, $f(x_k + c_k\Delta_k)$, $f(x_k - c_k\Delta_k)$ are the cost function values evaluated with a different sign of the perturbation. According to [8], the values of a_k and c_k are given by (6), where $A,\ a,\ c > 0$, using $\alpha = 0.602$, $\gamma = 0.101$ as suggested by [8].

$$a_k = \frac{a}{(k+1+A)^\alpha} \quad (5)$$

$$c_k = \frac{c}{(k+1)^\gamma}. \quad (6)$$

The traditional SPSA [8] calculates for each iteration k the simultaneous perturbation vector $\Delta_k n$ and performs two evaluations $f(x_k + c_k\Delta_k)$, $f(x_k - c_k\Delta_k)$ of the cost function (4) to estimate the gradient and update the optimization parameters. However, the parallel implementation proposed by [14] uses i slave processes, each one with its own perturbation vector Δ_k to increase the number of cost function evaluations per iteration. Thus, for each i process, the values of $f(x_k + c_k\Delta_k)$ and $f(x_k - c_k\Delta_k)$ are calculated. These gradients are combined with the one obtained by the master process to find the new direction of the gradient using (7), where $u(k-1,i)$ is a subscript that means the i-th process in the $k-1$ iteration. Thus, the next system input parameters x_{k+1} are calculated. Algorithm 1 summarize the process of the parallel SPSA algorithm.

$$d_k = \bar{g}_k + \frac{\bar{g}_k^T \bar{g}_{u_{(k-1,i)}}}{||\bar{g}_{u(k-1,i)}||^2} \bar{g}_{u_{(k-1,i)}} \quad (7)$$

$$x_{k+1} = x_k - a_k d_k. \quad (8)$$

Algorithm 1 Parallel SPSA Algorithm [14]

Input: a, c, A, α, γ
Output: x

Initialization : Initialize the SOC parallel framework and choose a, c, A, α, γ

1: Generate the Simultaneous perturbation vector Δ_k for each process i.
2: Calculate the approximated gradient using (4) Δ_k for each process i.
3: Choice of the combined gradient direction based on the norm $||\bar{g}_k(k-1,i)||^2$ using (7).
4: Update x_k applying (8)
5: Repeat 1 to 5 during m iterations
6: **return** x

III. Case study: Parallel Self Optimizing Control Framework for thermal systems

The parallel SOC framework is tested for the control of a thermal system. In this paper, the case study is the Digital Twin of a Peltier thermoelectric system proposed by [15]. As shown in Fig. 2, the system is composed of a Peltier heating cell (M1), a thermal infrared camera (M2) as a temperature sensor for uniformity temperature control, a LattePanda edge computer (M3), a power driver controlled by an Arduino (M4), and a battery (M5). In this case, the uniform temperature control system is closed-loop stable, employing a PI controller (9)

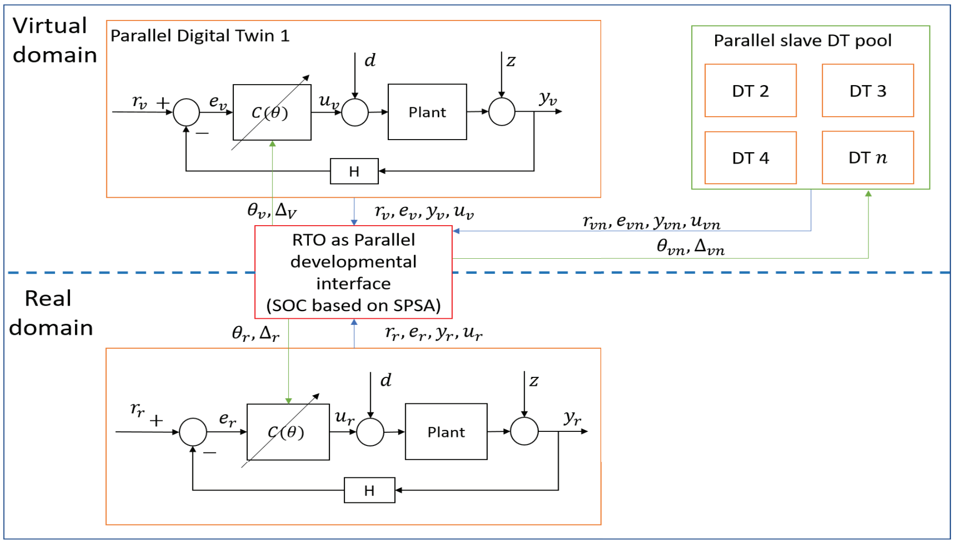

Figure 1: Parallel Self Optimizing Control Framework

Figure 2: DT case study: real-time vision feedback infrared temperature uniformity control

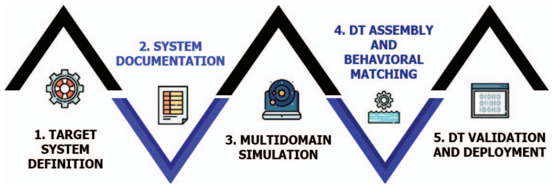

Figure 3: Five steps Digital Twin development framework

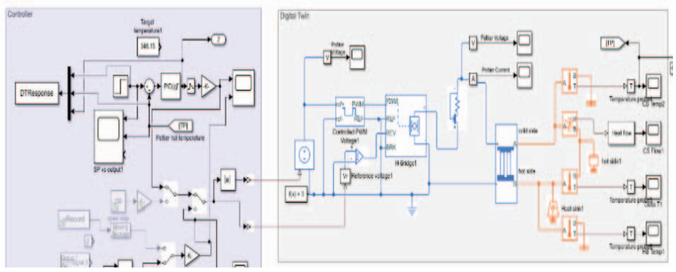

Figure 4: Peltier thermal system Digital Twin

with antiwindup, with the proportional and integral gains K_p and K_i as tuning parameters for the controller.

$$c(s) = k_p + \frac{k_i}{s}. \tag{9}$$

The Digital Twin of the thermal system is developed following the five steps Digital Twin systematic framework shown in Fig. 3. The DT uses Matlab/Simulink and Simscape to replicate the physical laws of the system and controls. The complete multiphysics simulation model is presented in Fig. 4. A detailed explanation of the Digital Twin development can be found in [15].

A. Parallel SOC framework evaluation

The parallel SOC framework is employed to optimize the proportional and integral gains of the PI controller (9) based on the cost function (1). For this purpose, two tests are proposed. The first test uses only the parallel interaction between the real system and the parallel Digital Twin 1 to optimize the values of the closed-loop PI controller employed in the system. Besides, the second test uses a pool of five Digital Twins of the system to accelerate the SPSA algorithm simultaneous perturbation with an independent perturbation vector Δ_k for each DT. Both tests are evaluated for a total of 200 iterations with the parameters $a = 60.17$, $\alpha = 0.602$, $\gamma = 0.101$, $c = 1.9$, and $W_1 = 1,\ W_2 = 0.1,\ W_3 = 1e-3$ for the SOC cost function (1) weights. Likewise, the initial conditions for the SPSA algorithm are given as a set of K_p, K_i that make the system stable obtained with the Ziegler-Nichols method [16]. For this reason, a First Order Plus Dead Time model of the system is identified using stepped inputs, which is given by (10), resulting in the initial values for K_p and K_i of $K_p = 10.3, K_i = 3.32$.

$$P(s) = \frac{2.7}{31.42s + 1} e^{-1.004s}. \tag{10}$$

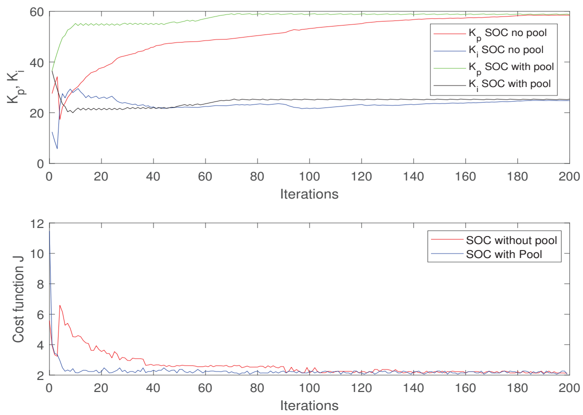

Figure 5: Parallel SOC test with real system and one mirror DT

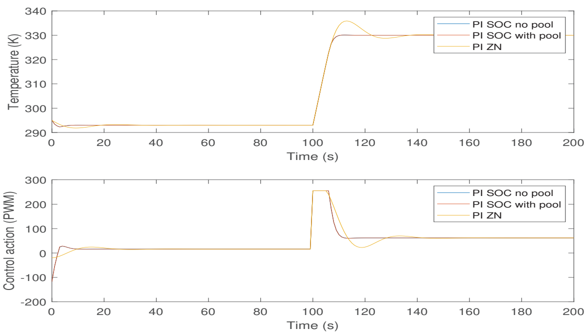

Figure 6: Parallel SOC test with real system, mirror DT and 10 DT slaves on pool

The cost function evaluation, the controller parameters evolution, and the time response of the optimized controllers for tests 1 and 2 are shown in Fig. 5 and Fig. 6. As can be observed, the parallel SOC with SPSA converges in both scenarios, with a convergence time of 180 iterations in test 1 (no DT pool), and 60 iterations when the DT pool is used. Likewise, the time response shows that the PI controller shows an improved performance after the self-optimization compared with the initial condition. Notice that the SOC control has been performed using the repetitive square reference signal shown in Fig. 6. Based on the obtained results, we can say that the parallel SOC framework with a Digital Twin pool can improve the closed-loop system response using an economical cost function based on the real-time updated system performance for each period.

IV. Conclusions and Future Works

This paper introduced a parallel SOC architecture supported by Digital Twin and the parallel SPSA algorithm. The Digital Twin of a thermoelectric system has been employed as a case study for the framework. The obtained results show that the parallel SOC control can significantly improve the system closed-loop performance, reducing the convergence time using parallel SPSA optimization supported by multiple instances of Digital Twin. Thus, the parallel SOC framework can be considered the initial step to introduce smartness into classic process control towards the implementation of smart control engineering. As future works, the convergence and stability analysis of the framework, its practical implementation using Hardware in the loop configuration, and its application for control problems with tens or hundreds of control variables to be tuned is proposed.

References

[1] S. Skogestad, "Plantwide control: the search for the self-optimizing control structure," *Journal of Process Control*, vol. 10, no. 5, pp. 487–507, 2000.

[2] K. Bariyur and M. Krstic, *Real-time optimization by extremum-seeking control*. Wiley-Interscience, 2003.

[3] H. S. Ahn, Y. Q. Chen, and K. L. Moore, "Iterative learning control: Brief survey and categorization," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1099–1121, 2007.

[4] Y. Sun, J. Reichelt, T. Bormann, and A. Gondorf, "A multi-step wafer-level run-to-run controller with sampled measurements for furnace deposition and CMP process flows: APC: Advanced process control," *2016 27th Annual SEMI Advanced Semiconductor Manufacturing Conference, ASMC 2016*, pp. 399–402, 2016.

[5] F. Y. Wang, L. Q. Yang, J. Yang, Y. Zhang, S. Han, and K. Zhao, "Urban intelligent parking system based on the parallel theory," *2016 International Conference on Computing, Networking and Communications, ICNC 2016*, 2016.

[6] S. Wang, J. Wang, X. Wang, T. Qiu, Y. Yuan, L. Ouyang, Y. Guo, and F. Y. Wang, "Blockchain-Powered Parallel Healthcare Systems Based on the ACP Approach," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 4, pp. 942–950, 2018.

[7] F. Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 630–638, 2010.

[8] J. Spall, *Introduction to stochastic search and optimization: Estimation, Simulation, and Control*. John Wiley & Sons, Incorporated, 1st ed., 2003.

[9] W. Ai, X. Li, and S. Tian, "A novel SPSA-based IMC-PID Data-driven Control Method," in *Chinese Control and Decision Conference (CCDC)*, pp. 4981–4986, IEEE, 2016.

[10] Priyatmadi, A. P. Sandiwan, H. Wijaya, and A. Cahyadi, "Application of SPSA LQR tuning on quadrotor," *Proceedings - 2016 6th International Annual Engineering Seminar, InAES 2016*, pp. 32–36, 2017.

[11] Y. F. Ning, W. S. Tang, and S. U. Lei, "Comparison between hybrid genetic-SPSA algorithm and GA for solving random fuzzy dependent-chance programming," *2005 International Conference on Machine Learning and Cybernetics, ICMLC 2005*, no. August, pp. 2742–2746, 2005.

[12] Z. Huajun, Z. Jin, and G. Tao, "Convergence accelerated by the improvements of stepsize and gradient in SPSA," *Proceedings of the 2011 Chinese Control and Decision Conference, CCDC 2011*, pp. 1–6, 2011.

[13] Y. Fan and T. Liu, "Parallel implementation of simultaneous perturbation stochastic approximation with adaptive step sizes," *ACM International Conference Proceeding Series*, pp. 3–6, 2018.

[14] H. Zhao and T. Liu, "A parallelized combined direction simultaneous perturbation stochastic approximation algorithm," *2017 2nd IEEE International Conference on Computational Intelligence and Applications, ICCIA 2017*, vol. 2017-January, no. 2, pp. 141–144, 2017.

[15] J. Viola and Y. Q. Chen, "Digital Twin Enabled Smart Control Engineering as an Industrial AI: A New Framework and Case Study," *2nd International Conference on Industrial Artificial Intelligence, IAI 2020*, 2020.

[16] D. Xue, Y. Chen, and D. Atherton, *Linear Feedback Control, Analysis and design with MATLAB - Advances in Design and Control*. SIAM, 2007.