Proceedings of the ASME 2025
International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC-CIE2025
August 17–20, 2025, Anaheim, CA

IDETC-CIE2025-168820

A DIGITAL TWIN BEHAVIORAL MATCHING FRAMEWORK VIA SPSA

Shiang Cao[1], Osama F. Abdel Aal[1], Marlon Romero[1], YangQuan Chen[1]

[1]Mechatronics, Embedded Systems and Automation (MESA) Lab,
Dept. of Mechanical Engineering,
University of California, Merced,
5200 N Lake Road, CA 95343, USA

ABSTRACT

Behavior matching is a critical process in aligning the dynamics of a physical system with those of its virtual replica. It plays a key role in the development of digital twins, as it directly impacts their accuracy and overall quality. In this study, we propose an online behavior matching framework based on simultaneous perturbation stochastic approximation (SPSA). By integrating an SPSA layer between the physical system and the digital twin model, the framework enables adaptive, real-time (or near real-time) tuning of the model parameters to improve matching accuracy. The feasibility and effectiveness of the proposed method are demonstrated through two case studies: behavior matching of a Simscape mass-spring-damper system and behavior matching of a DC motor. In both cases, the digital twin models successfully track the system's output, validating the proposed approach.

Keywords: Digital Twins, Behavior Matching, SPSA

1. INTRODUCTION

Digital twins, defined as a virtual representation of the real system[1], have garnered significant attention from both academia and industry. More than just high-fidelity simulations, digital twins can interact with their physical assets and are capable of reflecting any changes that occur to those assets[2]. As sophisticated computational frameworks, digital twins integrate data from sensors and other inputs to create accurate and adaptive simulations of complex systems [3, 4]. Due to these properties, Digital twins hold immense potential across a wide range of fields. In smart agriculture, digital twin frameworks have been proposed to enhance monitoring, productivity, and operational efficiency [5–7]. In mechatronic systems, digital twin models can be utilized for lifecycle analysis, predictive maintenance, and controller design [8–10]. In [11], a comprehensive five-step Digital Twin development framework is introduced. Within this framework, a key process known as behavioral matching (BM) is performed to tune the parameters of each subsystem that comprises the digital twin. This process ensures that the digital twin accurately mirrors the system's static and dynamic behaviors, maintaining alignment with its physical asset in the real world [10]. A description of the behavioral matching is presented in Fig. 1. As shown, the input and output data of the real system are fed into the digital model. An optimization loop is then implemented to identify the optimal parameters that minimize the mismatch error between the physical and digital systems. However, the conventional approach for behavioral matching still relies on collecting offline data and solving optimization problems based on various criteria, such as ISE, IAE, and ITAE [10–12]. In real-world scenarios, the dynamics of physical systems often vary during operation, while digital twins are expected to operate simultaneously with their physical counterparts in real time. Therefore, it is essential to develop a behavioral matching framework that can be executed in real or near-real time.

In this work, we present an operational behavioral matching method by introducing a Simultaneous Perturbation Stochastic Approximation (SPSA) layer between the physical systems and

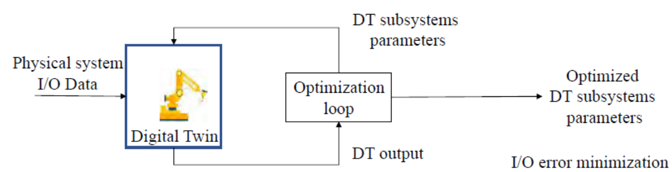


FIGURE 1: BEHAVIORAL MATCHING [11]

Corresponding author: YangQuan Chen. ychen53@ucmerced.edu and +1-209-2284672. The authors are all supported by the Center for Methane Emission Research and Innovation (CMERI) through the Climate Action Seed Funds grant (2023-2026) at the University of California, Merced. mechatronics.ucmerced.edu
Documentation for `asmeconf.cls`: Version 1.40, May 19, 2025.

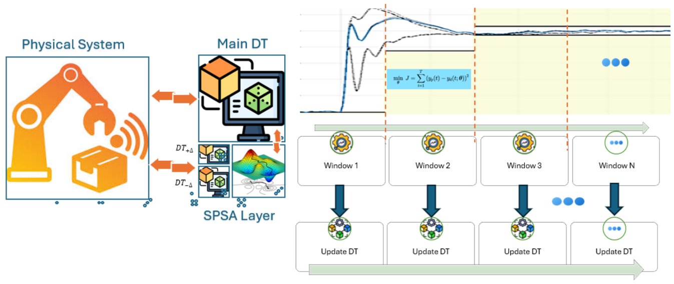


FIGURE 2: SPSA-BASED BEHAVIOR MATCHING FRAMEWORK

the digital twin models. Within this layer, the SPSA algorithm applies perturbations to the digital twin models and estimates the gradient of the matching error with respect to the parameters designated for tuning. Using the estimated gradient, the parameters of the digital twin models are iteratively updated at each time window, without affecting the continuous operation of the physical system or the digital twins.

2. SIMULTANEOUS PERTURBATION STOCHASTIC APPROXIMATION (SPSA) OPTIMIZATION

Simultaneous Perturbation Stochastic Approximation (SPSA) is a widely used zero-order optimization method that does not require explicit gradient information. It is particularly well-suited for scenarios where the objective function is computationally expensive or analytically intractable. Due to its efficiency in estimating/approximating the gradient, the SPSA technique is effective for high-dimensional optimization problems [13, 14]. SPSA has been successfully applied in various fields, including control systems, machine learning, and parameter estimation [15–17]. These characteristics make SPSA appropriate for behavioral matching frameworks in digital twins, where numerous parameters must be optimized to align the digital model's behavior with that of the physical system. An additional advantage of SPSA is its robustness to noise in the cost function, which further enhances its practical applicability.

In the SPSA framework, the gradient is approximated using only two evaluations of the defined cost function, denoted as $J(\theta)$ per iteration, regardless of the dimensionality of the system parameter θ being optimized. The iterative update rule is given as:

$$\theta_{k+1} = \theta_k - a_k \hat{g}_k, \tag{1}$$

where θ_k is the parameter vector in iteration k, a_k is the step size sequence, and $\hat{g}_k$ is the gradient estimate. The approximate gradient $\hat{g}_k$ is calculated using a simultaneous perturbation approach. Let Δ_k be a random perturbation vector, where each element $\Delta_{k,i}$ is independently drawn from a symmetric Bernoulli distribution (± 1 with equal probability). The gradient estimate is given by:

$$\hat{g}_k = \frac{J(\theta_k + c_k \Delta_k) - J(\theta_k - c_k \Delta_k)}{2c_k} \Delta_k^{-1}, \tag{2}$$

where c_k is the perturbation size, and Δ_k^{-1} represents element-wise inversion of Δ_k. The convergence of SPSA depends on

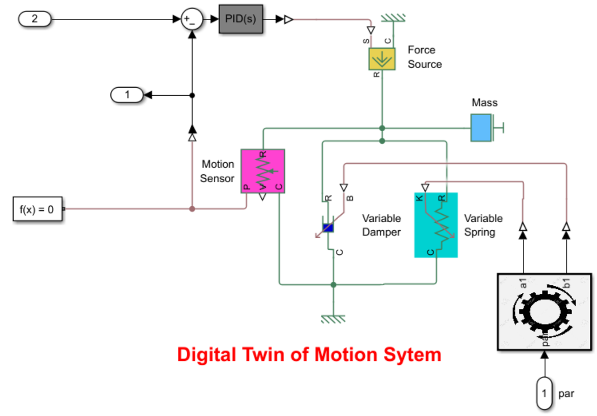


FIGURE 3: SIMSCAP MODEL OF THE MSD SYSTEM

properly choosing the sequences a_k and c_k, which typically follow the forms [14]:

$$a_k = \frac{a}{(A + k + 1)^{\alpha}}, \quad c_k = \frac{c}{(k+1)^{\gamma}}, \tag{3}$$

where a, c, A, α, γ are positive constants, and $\alpha \in (0.5, 1]$, $\gamma \in (0.5, 1]$. Under standard conditions on the step size and perturbation sequences, SPSA converges to a local minimum of the objective function [13, 18]. The asymptotic convergence rate is comparable to that of other stochastic gradient-based methods while maintaining significant computational advantages.

3. BEHAVIOR MATCHING FRAMEWORK AND RESULTS

The proposed behavior matching framework is illustrated in Fig. 2. During system operation, the total time period is divided into multiple segments or time windows. In each time window, a cost function is evaluated to quantify the difference in response between the real system and a set of perturbed digital twin (DT) models. At the end of each window, this information is sent to the SPSA optimizer, which approximates the gradient and generates updated parameters to refine the main DT model and create new perturbed DT models for the next iteration. This process enables continuous refinement of the DT in real or near-real time.

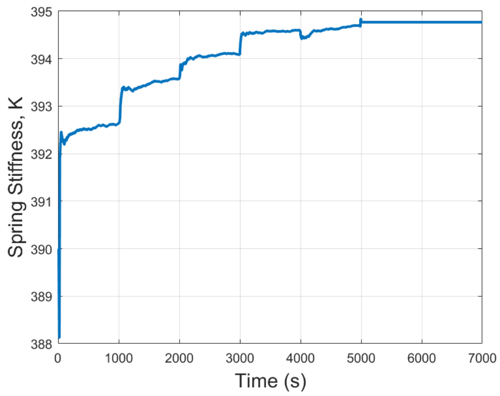
3.1. Case Study: Mass-Spring-Damper System

In this section, a physical modeling example is discussed. Figure 3 illustrates the block diagram of the mass-spring-damper (MSD) system. The system is governed by the following second-order differential equation:

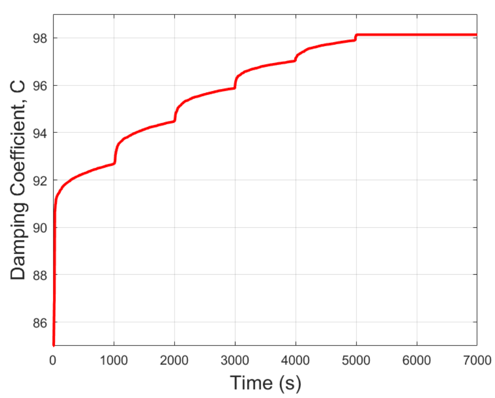
$$m\ddot{x} + c\dot{x} + kx = F,$$

where x denotes the displacement, F is the external force, and m, c, and k represent the mass, damping coefficient, and stiffness, respectively. In this case study, the stiffness k and damping coefficient c are treated as tunable parameters and are adjusted through the SPSA layer to match the behavior of the digital twin model with that of the physical system.

In the simulation, the input reference position $r(t)$ is a square wave ranging from 2 to -2, with a period of $T = 20s$ and a duty



(a) Trajectory of the stiffness *k*



(b) Trajectory of the damping coefficient *c*

FIGURE 4: TRAJECTORIES OF THE MSD SYSTEM'S PARAMETERS

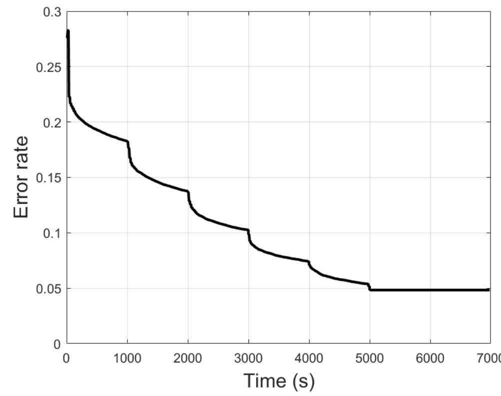
cycle of 50%. According to [14], the parameters of the SPSA layer are set as $a = 200, c = 90, \alpha = 0.602, \gamma = 0.101$, and $A = 1$. The initial stiffness and damping coefficient are set to 390 and 85, respectively, while the true values are 400 and 100. The SPSA layer updates the parameters within a time window of $T_1 = 10s$. To quantify the behavior matching performance, the residual error rate is defined as

$$e(n) = \frac{\int_{nT}^{(n+1)T} |y_m - y| dt}{\int_{nT}^{(n+1)T} |y| dt} \times 100\%,$$

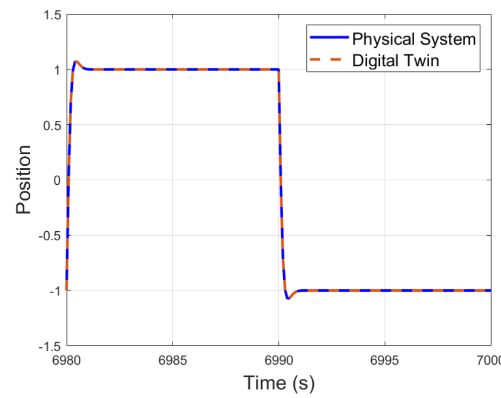
where $n \in \mathbb{N}_+$, and y_m, y are the outputs from the DT model and the real system, respectively. The SPSA layer terminates when the residual error rate reaches or falls below the threshold of 0.05%. To accelerate the learning speed, the iteration count is reset to 1 if it exceeds 100 iterations. As shown in Fig. 4, the estimated values of stiffness k and damping coefficient c gradually approach the true values. The results in Figure 5 further demonstrate that the residual error decreases successfully to the specified threshold, and the digital twin model's output closely matches that of the physical system.

3.2. Case Study: DC Motor

This section presents a case study on the velocity behavior matching for a DC motor. As illustrated in Fig. 6, the system



(a) The error rate of the behavior matching



(b) Outputs of the DT model and real system

FIGURE 5: RESULTS OF THE BEHAVIOR MATCHING

consists of a $12V$, 130RPM DC motor equipped with a Hall-effect encoder, a motor shield, and a microcontroller that sends control commands and communicates with the DT model. The DT model representing the motor's angular velocity $w(t)$, given an input voltage $v(t)$, is modeled using a second-order transfer function:

$$G(s) = \frac{c}{s^2 + as + b},$$

where a, b, and c are system parameters tuned to match the DT model's velocity behavior with that of the physical system. To reduce measurement noise, the output velocity is processed using

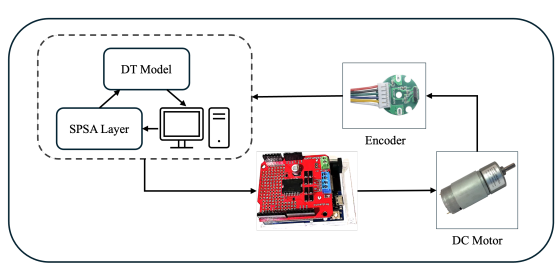
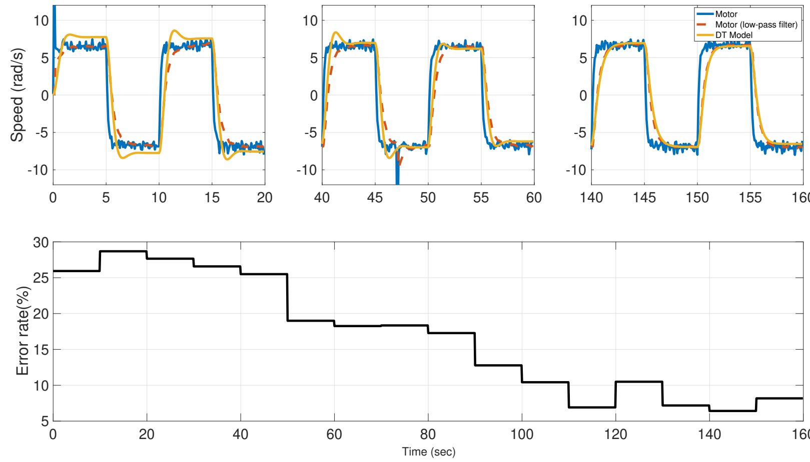
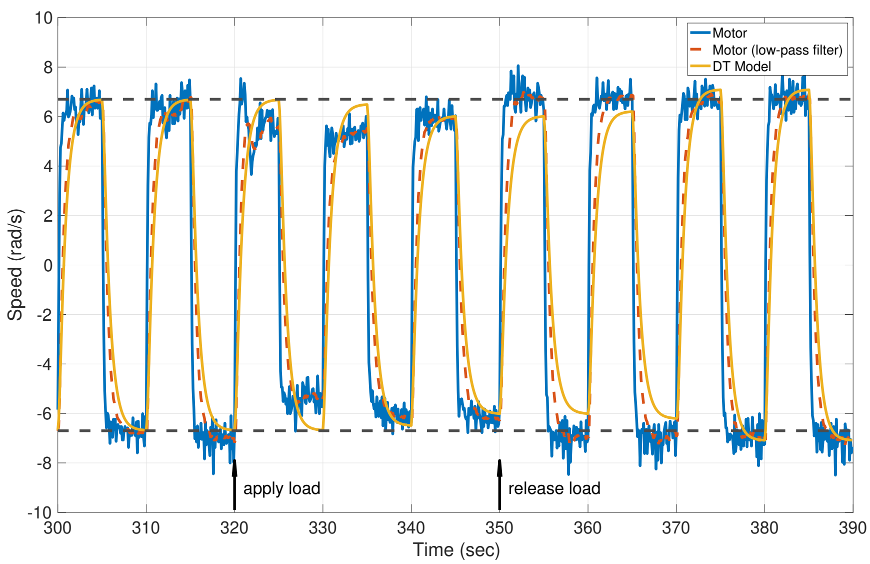


FIGURE 6: BLOCK DIAGRAM OF BEHAVIOR MATCHING OF THE DC MOTOR



(a) First experiment: Actual speed, DT output, filtered speed, and error rate



(b) Second experiment: Actual speed, DT output, and filtered speed

FIGURE 7: EXPERIMENT RESULTS IN THE DC MOTOR CASE STUDY.

a first-order low-pass filter:

$$H(s) = \frac{1}{\tau s + 1},$$

where τ is the time constant, set to $0.5s$.

In the first experiment, the input u(t) is a square wave ranging from 2 to -2 (with the negative value indicating a reversal in direction), having a period of $10s$ and a 50% duty cycle. The parameters of the SPSA layer are set as $a = 2, c = 0.1, \alpha = 0.602, \gamma = 0.101$, and $A = 100$. The initial parameters are perturbed by 10% from the values obtained through system identification, and the SPSA layer updates the parameters over a time window of $T_1 = 10s$. From Fig. 7a, the output from the DT model initially deviates from the system output but gradually converges toward it. After approximately 140 seconds, the DT model output closely matches the filtered output of the physical motor.

In the second experiment, external disturbances are introduced during motor operation. As shown in Fig. 7b, a payload is applied to the motor at $t = 325s$. Despite the disturbance, the DT model continues to track the motor's output. After the load is removed at $t = 350s$, the DT model maintains tracking accuracy and gradually recovers to reflect the system's original output behavior. This demonstrates the model's ability to adapt to varying conditions during operation.

4. CONCLUSION AND FUTURE WORK

The proposed online behavior matching framework enables real-time adjustment of the digital twin (DT) model's parameters during system operation. This allows the DT to accurately track the physical system, which is inherently time-varying and subject to environmental disturbances. Two case studies have demonstrated the effectiveness of the framework. In particular, the second case study shows that the DT model can successfully adapt its parameters in response to changes in the operating conditions of the physical system. This adaptive capability suggests the potential for using DT parameter updates to infer the condition of the physical system, enabling applications such as predictive maintenance and system health monitoring. In future work, the framework will be extended to more complex systems with multiple interacting subsystems and tested under a broader range of real-world operating conditions.

ACKNOWLEDGMENT

Y. Chen and S. Cao acknowledge the financial support from NSF grant CBET-1856112 under the award entitled "INFEWS:T2: Saltwater Greenhouse System for Agricultural Drainage Treatment and Food Production" and Climate Action Seed Grant on Center for Methane Emission Research and Innovation (CMERI) at UC Merced. S. Cao was also supported in part by an F3 R&D GSR Award (Farms Food Future Innovation Initiative (or F3), as funded by the US Dept. of Commerce, Economic Development Administration Build Back Better Regional Challenge).

REFERENCES

[1] Grieves, Michael. "Digital twin: manufacturing excellence through virtual factory replication." *White paper* Vol. 1 No. 2014 (2014): pp. 1–7.

[2] Viola, Jairo and Chen, YangQuan. *Digital Twin Background*. Springer International Publishing (2023): pp. 1–21.

[3] Viola, Jairo, Guc, Furkan and Chen, YangQuan. "Digital Twins–Enabled Smart Control Engineering and Smart Predictive Maintenance." *Handbook of Digital Twins*. CRC Press (2024): pp. 584–600.

[4] Mohammed, Wael M. *A Methodology for Architecting Digital Twins in Factory Automation*. Tampere University Dissertations - Tampereen yliopiston väitöskirjat, Tampere University (2024).

[5] Verdouw, Cor and Kruize, Jan Willem. "Digital twins in farm management: illustrations from the FIWARE accelerators SmartAgriFood and Fractals." *in Proceedings of the 7th Asian-Australasian Conference on Precision Agriculture Digital, Hamilton, New Zealand*: pp. 16–18. 2017.

[6] Chaux, Jesus David, Sanchez-Londono, David and Barbieri, Giacomo. "A digital twin architecture to optimize productivity within controlled environment agriculture." *Applied Sciences* Vol. 11 No. 19 (2021): p. 8875.

[7] Cao, Shiang and Chen, YangQuan. "A Digital Twin Development Framework for a Smart Saltwater Greenhouse." *in Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 87356: p. V007T07A004. 2023. American Society of Mechanical Engineers.

[8] Tuegel, Eric J, Ingraffea, Anthony R, Eason, Thomas G and Spottswood, S Michael. "Reengineering aircraft structural life prediction using a digital twin." *International Journal of Aerospace Engineering* Vol. 2011 No. 1 (2011): p. 154798.

[9] Guc, Furkan. "Smart Predictive Maintenance Enabled by Digital Twins and Physics Informed Smart Big Data." Ph.D. Thesis, eScholarship, University of California. 2023.

[10] Viola, Jairo. "Self-Optimizing Smart Control Engineering Enabled by Digital Twins." Ph.D. Thesis, University of California, Merced. 2022.

[11] Viola, Jairo and Chen, YangQuan. *A Digital Twin Development Framework*. Springer International Publishing (2023): pp. 23–45.

[12] Nwoke, Justus, Milanesi, Marco, Viola, Jairo and Chen, YangQuan. "FPGA-based digital twin implementation for power converter system monitoring." *in Proceedings of the 2023 IEEE 3rd International Conference on Digital Twins and Parallel Intelligence (DTPI)*: pp. 1–6. 2023. IEEE.

[13] Spall, James C. *Introduction to stochastic search and optimization: estimation, simulation, and control*. John Wiley & Sons (2005).

[14] Spall, James C. "Implementation of the simultaneous perturbation algorithm for stochastic optimization." *IEEE Transactions on Aerospace and Electronic Systems* Vol. 34 No. 3 (1998): pp. 817–823.

[15] Zhou, Ya-Li, Zhang, Qi-Zhi, Li, Xiao-Dong and Gan, Woon-Seng. "On the use of an SPSA-based model-free feedback controller in active noise control for periodic disturbances in a duct." *Journal of Sound and Vibration* Vol. 317 No. 3-5 (2008): pp. 456–472.

[16] Song, Qing, Spall, James C, Soh, Yeng Chai and Ni, Jie. "Robust neural network tracking controller using simultaneous perturbation stochastic approximation." *IEEE Transactions on Neural Networks* Vol. 19 No. 5 (2008): pp. 817–835.

[17] Yuan, QingHui. "A model free automatic tuning method for a restricted structured controller by using simultaneous perturbation stochastic approximation (spsa)." *in Proceedings of the 2008 American Control Conference*: pp. 1539–1545. 2008. IEEE.

[18] Spall, James C. "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation." *IEEE Transactions on Automatic Control* Vol. 37 No. 3 (1992): pp. 332–341.