Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

OPTIMIZING DIGITAL TWIN SYNCHRONIZATION IN A FINITE HORIZON

Barış Tan

College of Administrative Sciences and Economics,
and College of Engineering
Koç University
Rumelifeneri Yolu, Sarıyer
Istanbul, 34450, TURKEY

Andrea Matta

Department of Mechanical Engineering
Politecnico di Milano
Via La Masa 1
Milano, 20156, ITALY

ABSTRACT

Given the tendency to increase the complexity of digital twins to capture a manufacturing system in the most detailed way, synchronizing and using a complex digital twin with the real-time data may require significant resources. We define the optimal synchronization problem to operate the digital twins in the most effective way by balancing the trade-off between improving the accuracy of the simulation prediction and using more resources. We formulate and solve the optimal synchronization problem for a special case. We analyze the characteristics of the state-dependent and state-independent optimal policies that indicate when to synchronize the simulation at each decision epoch. Our numerical experiments show that the number of synchronizations decreases with the synchronization cost and with the system variability. Furthermore, a lower average number of synchronizations can be achieved by using a state-dependent policy.

1 INTRODUCTION

Digital twins are considered as integral parts of smart manufacturing and Industry 4.0 (Tao, Qi, Wang, and Nee 2019). In parallel to the increase in developing and using digital twins in manufacturing, the market size of digital twin software reached 2.26 billion USD in 2017 and it is expected to reach 26.7 billion USD in 2025 with an annual growth of 38% (Grandview Research 2021).

With the advances in data-collection and processing technology, there is a continuing trend to increase the detail of digital twins to capture the behaviour of a complex system in the most realistic and detailed way. A digital twin of a complex system such as a manufacturing line or a large automated warehouse may include thousands of variables that track the dynamics of the system. Increasing the complexity of a digital twin introduces challenges for maintaining, synchronizing, and using digital twins effectively. More specifically, synchronizing all the variables with the data collected from the system takes time, uses resources, and may hinder real-time operation of a digital twin. Indeed, a digital twin not aligned with the actual state of the physical system might provide inaccurate predictions. Several industrial and service applications of digital twins will be related to this problem. For instance, a production line that observes an important downtime at one of the machines in a station with parallel machines will have a different production output until the failure is fixed. If the digital twin is used as simulation to predict the daily production, a new simulation experiment must be executed after having initialized the simulation model with the new state of the real system (e.g, the failed machine cannot work and it is temporally replaced with an old and slower machine).

Therefore, an operational decision needs to be made to determine when and how to synchronize a digital twin. We define and analyze the optimal synchronization problem as a dynamic control problem. The objective of the control problem is to operate the digital twins in the most effective way by balancing the benefits of increasing their accuracy by more frequent synchronization actions with the alignment costs

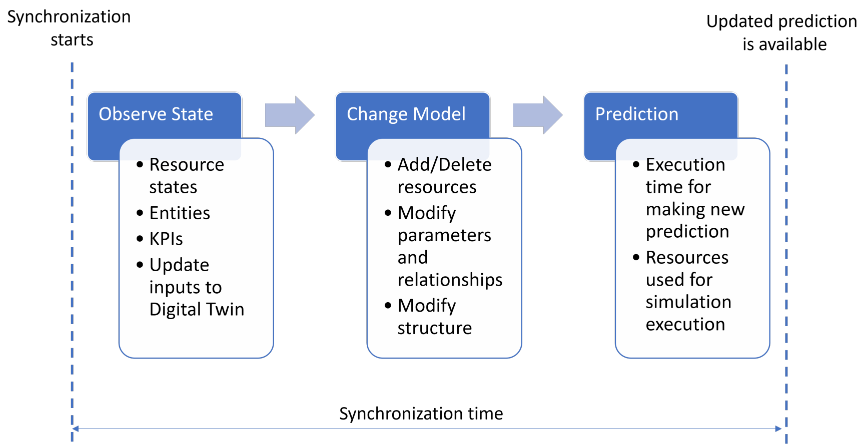

Figure 1: Main phases and activities of a synchronization.

to be incurred. The alignment costs are related to the resources used to observe the state from the physical system, gather position of entities, data and KPI values, and input them into the digital twin. It may also happen that the new physical system state was unforeseen and this may force a change of the simulation model too. Last, the time and resources used to make a new prediction can also be important in case of complex systems and very time-consuming simulations such as in semiconductor manufacturing. Figure 1 graphically summarizes the main phases that can be executed during a synchronization as well as the activities potentially contributing to synchronization costs.

The synchronization problem is not strictly limited to digital twins, but it also occurs when traditional simulation is used for predictions. For instance, in large manufacturing job shops, the scheduling and rescheduling of jobs is fundamental to keep the production under control and meet the due dates, particularly when disruptions occur. In many cases, the scheduling plan is generated using predictions from a discrete event simulation model that needs the current state of resources and jobs as an input every time a new plan must be generated. In fact, in many industrial plants the collection of the whole job shop state can be very expensive in time and human resources, and also for this reason rescheduling is not frequent.

In the literature, there are many studies and surveys related to discussing digital twins (Kritzinger, Karner, Traar, Henjes, and Sihn 2018), (Uhlemann, Lehmann, and Steinhilper 2017), (Barricelli, Casiraghi, and Fogli 2019). However, the number of studies that focus on optimizing the operation of a digital twin is limited. When digital twins are used as a forecasting tool for production planning, e.g (Cardin and Castagna 2011), different operational problems arise. For example, Hanisch, Tolujew, and Schulze (2005) define the problem of initialization of online simulation models that require mapping system variables with the available data. In the context of balancing the cost of model validation with the expected benefit from the model, Sargent (2010) states that the value of a model increases with the confidence in the model at a decreasing rate. In the context of digital twins, the confidence of a simulation model will be higher if its parameters are updated more frequently. However, collecting the data from a real system and updating the parameters of a simulation come at a cost that is related to possible delays in the process. As the simulation model gets closer to the physical system, it gets more complicated and the number of variables to be updated at each synchronization point also increases.

In this study, we define the problem of optimizing the synchronization intervals for a digital twin. The basic question we focus is when the digital twin should be synchronized in order to manage the trade-off between improving the accuracy of the simulation, i.e., reducing the bias, and therefore minimize the costs related to inaccurate predictions and use of resources in the most balanced way.

The organization of the remaining part of the paper is as follows. Section 2 introduces the model and its assumptions and state the optimal synchronization problem formally. Section 3 presents the mathematical

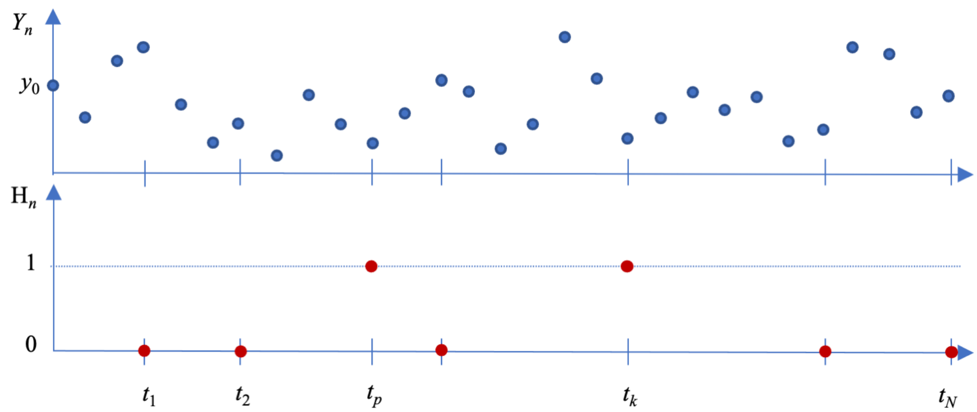

Figure 2: Sample realization of a simulation tracking Y_N with the synchronization (H_n) decisions given at different decision epochs.

programming formulations for the state-independent and state-dependent optimal synchronization problems for a special case. The results of the numerical experiments are given in Section 4. Finally, the conclusions are given in Section 5.

2 OPTIMAL DIGITAL TWIN SYNCHRONIZATION PROBLEM

The objective of the optimal digital twin synchronization problem is finding an optimal synchronization policy that indicates whether to synchronize the simulation dynamically at each decision epoch based on the data collected from the system. Figure 2 shows the evolution of a performance measure tracked by a simulation (Y_n) and the synchronization ($\mathrm{H}_n \in \{0,1\}$) decisions given at different decision epochs. In this example, the digital twin is synchronized at decision epochs t_p and t_k and not synchronized at other epochs. The formal definition of the optimal synchronization problem is presented next following the description of the physical system and its digital twin.

2.1 Physical system description

A real system operates with stochastic behavior and evolves with time epochs $k = 0,1,\ldots,n,\ldots$. The real system is assumed to be stationary, i.e., its statistical properties do not change over time. The system performance measure at time t_k is a random variable Y_k. Assuming we are at time epoch n, we are interested in predicting the expected performance at time τ, $\mathbb{E}[Y_\tau]$ within the time interval $n+1 \leq \tau \leq n+\delta$ as well as the expectation of other functions that depend on the system performance measure at time τ. The parameter δ is given and can be considered as the forecasting horizon. The tuple of observations available at time t_n with the last observation at time t_k, $t_k < t_n$ is depicted with the tuple $\mathbf{y}_{n,k} = (y_0, y_1, \ldots, y_k)$. Then the full history at time t_n is $\mathbf{y}_{n,n}$. The expected value of the performance measure given the full history is $\mathbb{E}[Y_\tau|\mathbf{y}_{n,n}]$.

2.2 Digital system description

A digital twin is available to numerically calculate $\hat{Y}_\tau$, i.e., the predicted system performance at time epoch τ. Due to the stochastic behavior of the physical system, it may happen that the digital twin gets misaligned from the physical system thus jeopardizing its prediction capabilities. This part presents the model of the digital twin used to decide when synchronizing digital and physical systems.

It is assumed that the both the simulation model and its input parameters are given. The former has been implemented according to the conceptual model of the physical system developed at a certain detail level d, where higher values correspond to more detailed models. Simulation input parameters are fitted based on real observations of the variables that are modeled as random variables in the simulation model.

Examples of such variables are part arrivals, failure events, processing times, etc. The set of the fitted input simulation parameters is denoted with $\mathbf{p}$. Both the detail level and the parameters of the simulation model can finely be calibrated by comparing the output performance of the simulation model $\hat{Y}_k$ with the real one Y_k after a large enough observation time, for this reason we will assume in the remainder of Section 2 that d and $\mathbf{p}$ are given.

2.2.1 States

The digital twin is a simulation program run at initial time 0 and evaluated at different integer time instances $t_1, t_2, \ldots, t_k, \ldots, t_n$. The state of the digital twin at time t_n, before the action at time t_n is taken, is $\mathbf{S}_n = (\mathbf{y}_{n,k})$ where $\mathbf{y}_{n,k}$ is the tuple of the system observations until the last synchronization epoch at time t_k. That is, with the last synchronization completed at time t_k (with $t_k < t_n$), the information available is $\mathbf{y}_{n,k} = (y_0, y_1, \ldots, y_k)$.

The performance of the system at time τ predicted by simulation is a function g that depends on the values of the performance measures last synchronized at time t_k, $t_k \leq t_n$, i.e., $\mathbf{y}_{n,k}$ and the prediction interval. This estimate is denoted with function $g(\tau, \mathbf{y}_{n,k}; d, \mathbf{p})$, $\mathbb{E}[\hat{Y}_\tau | \mathbf{S}_n] = g(\tau, \mathbf{y}_{n,k}; d, \mathbf{p})$ to make explicit that τ and $\mathbf{y}_{n,k}$ are the argument whereas d and $\mathbf{p}$ are given values.

Since the digital twin is not a perfect representation of the real system we can write in general that $\mathbb{E}[\hat{Y}_\tau | \mathbf{S}_n] \neq \mathbb{E}[Y_\tau | \mathbf{y}_{n,n}]$. Another cause of the prediction error is not being able to use all the information available on the true performance until time t_n. Indeed, as the digital twin uses the new information, the prediction is expected to improve. Therefore, in order to improve the accuracy of performance prediction made at time t_n, it is possible to synchronize simulation with the real system (i.e.,, collecting y_n and using $\mathbf{y}_{n,n}$ to make the prediction).

2.3 Actions

Given that the system has been synchronized at time t_k, at time $t_n > t_k$, different decisions can be given to increase the detail level of the simulation, synchronize the simulation or update the parameters of the simulation. In this study, we assume that the detail level and the system parameters are given. Therefore, the only action is deciding whether or not synchronizing the digital twin with the physical system. The decision at time t_n is given by $\mathbf{u}_n = (\mathrm{H}_n)$ where H_n is the synchronization decision. Taking the decision is indicated by 1 and not taking the decision is indicated by 0. The set of decisions taken from time t_0 until t_n is given in the tuple $\chi_n = (\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{n-1})$.

If a **synchronization** action is not taken at time t_n, the state of the digital twin does not change, i.e., $\mathbf{S}_{n+1} = \mathbf{S}_n = \mathbf{y}_{n,k} = (y_0, y_1, \ldots, y_k)$. The synchronization action at time t_n adds all the observations since the last synchronization of the system at time t_k, $(y_{k+1}, y_{k+2}, \ldots, y_n)$ to the synchronization history. That is $\mathbf{S}_n = \mathbf{y}_{n,n} = (y_0, y_1, \ldots, y_n)$ is used to make the prediction. Since the additional information can only be used to improve the accuracy and is ignored otherwise, the accuracy of the system cannot degrade with a synchronization action. With this observation, the prediction of the expected value of the performance measure at time τ, $n+1 \leq \tau \leq n+\delta$ can be updated and the bias of the estimated performance will likely be reduced. However, this reduction is likely to decrease as τ increases.

The state $\mathbf{S}_{n+1}$ at time t_{n+1} is determined based on the state at previous time $\mathbf{S}_n$ and on the action $\mathbf{u}_n = (\mathrm{H}_n)$ taken at time t_n:

$$\mathbf{S}_{n+1} = S^{\mathrm{M}}(\mathbf{S}_n, \mathbf{u}_n) = \begin{cases} (y_0, y_1, \ldots, y_k, \ldots, y_n) & \mathrm{H}_n = 1 \\ (y_0, y_1, \ldots, y_k) & \mathrm{H}_n = 0 \end{cases}, \tag{1}$$

where S^{M} denotes the transition function.

The real system operates with stochastic behavior and evolves with time epochs $k = 0, 1, \ldots, n, \ldots, N$. Assuming we are at time epoch n, we are interested in predicting the expected performance at time τ, $\mathbb{E}[Y_\tau]$ within the time interval $n+1 \leq \tau \leq N$ as well as the expectation of other functions that depend on the system performance measure at time τ.

2.4 Costs

Each synchronization action has a cost that can be conceived proportional to the time and resources used to align the digital twin with its physical counterpart. The synchronization action cost is c_{H} and the total digital twin cost of action $\mathbf{u}_n = (\mathrm{H}_n)$, is given as

$$C_{\mathrm{DT}}(\mathbf{u}_n) = c_{\mathrm{H}}\mathrm{H}_n. \tag{2}$$

The cost related to the prediction bias for the prediction at time τ, denoted with $c_{\mathrm{B}}(\tau, \mathbf{S}_n, \mathbf{u}_n)$ is dependent on the difference between the expected value of the performance measure at time τ (with $n+1 \leq \tau \leq N$) obtained by using the full history at time t_n and the digital twin prediction by using the history that depends on the synchronization decision at time t_n that determines the history of observations, $\mathbf{y}_{n,k}$. The prediction bias cost related to the synchronization decision at time t_n for the period N is assumed as follows:

$$C_{\mathrm{B}}(N, \mathbf{S}_n, \mathbf{u}_n) = h\left(\mathbb{E}[Y_N|\mathbf{y}_n], g(N, \mathbf{y}_{n,k}; d, \mathbf{p})\right), \tag{3}$$

where function $h(\cdot)$ is a convex function. The synchronization action cost can be set depending on the relative accuracy improvement to be obtained with a synchronization action that requires a particular level of resource to be used.

2.5 THE OPTIMAL SYNCHRONIZATION PROBLEM

The main problem investigated in this work is deciding to synchronize or not the digital twin with the real system. The decision can be taken at any time epoch until N and balances the trade–off between the bias of the prediction and the total costs incurred to make the prediction. If no action is taken, the cost will be null but the bias will be large. On the contrary, if we frequently synchronize the digital twin, the cost will be large with the benefit of small bias. The problem can be formulated as a mathematical program under the full information or as a sequential decision making problem.

2.5.1 Mathematical Programming Formulation

In order to analyze the properties of the optimal synchronization and updating policy with the full information about the dynamics of the performance measure of the physical system, the following mathematical programming must be solved to determine the decision values $\mathbf{u}_n$, $n = 0, 1, \ldots$:

$$\min_{\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_N} \quad \sum_{n=0}^{N} C_{\mathrm{DT}}(\mathbf{u}_n) + \mathbb{E}[C_{\mathrm{B}}(S^{\mathrm{M}}(\mathbf{S}_n, \mathbf{u}_n), \mathbf{u}_n))|\mathbf{S}_n], \tag{4}$$

where the cost functions are defined with Equations (2) and (3), subject to Equation (1). The stochastic process that governs the performance measure of the physical system is used to calculate the expectation given in Equation (4). In order to solve this problem, the cost functions need to be evaluated based on the action taken and their effect on the prediction accuracy of the digital twin.

Once these specific mechanisms and the stochastic process of the performance measure of the physical system are given, the optimization problem given in Equation (4) is a convex nonlinear integer programming formulation. The solution of this problem can be used to analyze the properties of the optimal synchronization policy with the full information.

2.5.2 Sequential Decision Making Problem

The above mathematical programming formulation can be used to determine the optimal policy with the full information. Without availability of full information, a real-time control policy that acts on the synchronization decisions at each epoch based on the real-time data collected from the system needs to be developed. This is a more challenging problem since the evolution of the performance measure of the physical system as well as the effect of the synchronization actions on the accuracy improvement need to be learned and updated with the new information starting with the prior estimates. This problem can be formulated as a sequential decision making/real-time stochastic control problem (Powell 2021).

In this setting, the objective is finding a policy Π that sets the decision variables $\mathbf{u}_n$ based on the observed state variables $\mathbf{S}_n$, i.e., $\mathbf{u}_n = U^{\Pi}(\mathbf{S}_n)$ to minimize the expected cumulative cost of prediction bias and of synchronizations:

$$\min_{\Pi} \quad \sum_{n=0}^{N} C_{\mathrm{DT}}(U^{\Pi}(\mathbf{S}_n)) + \mathbb{E}[C_{\mathrm{B}}(S^{\mathrm{M}}(\mathbf{S}_n, U^{\Pi}(\mathbf{S}_n))|\mathbf{S}_0], \tag{5}$$

where the cost functions are defined with Equations (2) and (3) and Equation (1) describes the transition equations. With the exogenous random process given for the evolution of the performance measure of the physical system, this problem can be analyzed as a sequential decision making problem by using different approaches including reinforcement learning.

3 ANALYSIS OF A SPECIAL CASE

In this section, we analyze a specific system to state the problem explicitly and investigate the optimal synchronization policy with full information. That is, the optimal decision is determined by having the full information about the probabilistic behavior of the underlying physical system. With the full information, there is no bias in prediction, i.e., $\mathbb{E}[\hat{Y}_\tau|\mathbf{y}_{n,n}] = \mathbb{E}[Y_\tau|\mathbf{y}_{n,n}]$. Therefore, this setting allows us to analyze the optimal policy without the effect of bias due to having limited information.

3.1 Problem Definition

In this study, we focus on the problem of predicting the number of successes in N trials with a success probability of p in each trial. Let Y_k be the number of successes in the kth trial. We are at trial n, we are interested in predicting $\mathbb{E}[Y_N]$. We can either synchronize, i.e., observe and use the number of successes at trial n in our prediction of $\mathbb{E}[Y_N]$ or use our previous observations to produce the prediction. Observing and using the number of successes at trial n will improve our predictions, but it will come at a cost. The optimal decision should balance the marginal benefit from synchronization with its cost to determine whether or not synchronize at time n. Despite the simplicity of this special case, it is representative of several problems such as the prediction of cumulative production in manufacturing shop floor or the prediction of delivered orders in service and production facilities.

In the case of synchronization, only the last observation of Y_n, not the history yields the best prediction:

$$\mathbb{E}[Y_N|\mathbf{y}_{n,n}] = \mathbb{E}[Y_N|y_n] = (N-n)p + y_n. \tag{6}$$

If the system is not synchronized and the prediction is based on a previously observed value y_k in trial k, $k < n$, then the estimate will be worse than the one given above

$$\mathbb{E}[\hat{Y}_N|\mathbf{S}_n] = \mathbb{E}[Y_\tau|y_k] = (N-k)p + y_k. \tag{7}$$

We use $C_{\mathrm{B}}(\mathbf{S}_n) = \left(\mathbb{E}[Y_N|\mathbf{y}_n] - \mathbb{E}[\hat{Y}_N|\mathbf{S}_n]\right)^2$ as the bias cost.

We consider two different policies: state-independent policy and state-dependent policy. In the state-independent policy, the synchronization decision is made without differentiating the state of the system $\mathbf{S}_n = (\mathbf{y}_{n,k})$. That is, the decision is given only based on n, the trial number. On the other hand, the state-dependent policy differentiates the decision based on the observation in trial n and also the history $(\mathbf{y}_{n,k})$.

3.2 State-Independent Policy Formulation

For the state independent case, the following formulation is used to determine the optimal state-independent policy, $\mathbf{u}_n = \mathrm{h}_n$, $n = 0, 1, \ldots, N$:

$$\min_{\{\mathrm{h}_n\}} \quad \sum_{n=1}^{N} \left(c_{\mathrm{H}} \mathrm{h}_n + \mathbb{E}\left[(EY_n - ES_n))^2 \right] \right) \tag{8}$$

$$EY_n = (N-n)p + Y_n, \qquad n = 1, \ldots, N \tag{9}$$

$$ES_n = ES_{n-1} + (EY_n - ES_{n-1})\mathrm{h}_n, \qquad n = 1, \ldots, N \tag{10}$$

$$ES_0 = Np, \tag{11}$$

$$\mathrm{h}_n \in \{0, 1\}, \qquad n = 1, \ldots, N. \tag{12}$$

In this formulation, the total cost of synchronization and the prediction bias cost is given as the objective function given in Equation (8), where EY_n and ES_n represent the true and digital twin predictions respectively. The state transition function given in Equation (1) that defines the next state depending on the decision is represented as Equation (10) in the formulation. Note that decision variables h_n only depend on the time epoch n and not on the system state. The initial prediction for the number of successes in N trials with a success probability of p is Np and it is given in Equation (11). The expression for the prediction of the expected total number of successes in N trials with the observation of the number of successes in trial n is given in Equation (9), which also includes the realization of the random variable $Y_n = y_n$. As a result, the formulation given in Equations (8)-(12) is a stochastic nonlinear integer programming formulation. For the general case, the solution of the above stochastic optimization problem can be determined by using a scenario approach or an expected value formulation.

3.2.1 Scenario Approach

In this section we formulate the state independent problem by enumerating all the possible scenarios and averaging the prediction cost. Specifically, the cost deriving from the decision taken at each period is averaged over all the possible combinations. For N periods, the number of all possible combinations is 2^N. This formulation is often called the deterministic equivalent linear program and can be afforded only for small values of N. Since the objective of this study is defining the synchronization problem and investigating the state-independent and state-dependent policies, the scenario approach is suitable by selecting N in an appropriate way.

In each scenario $j = 1, \ldots, 2^N$, the digital twin prediction can be updated upon a synchronization (i.e.,, $ES_{n,j} = EY_{n,j}$) or taken from the previous period (i.e.,, $ES_{n,j} = ES_{n-1,j}$) of the same scenario j. Let $R = r_{n,j}$, $n = 1, \ldots, N$, $j = 1, \ldots, 2^N$, be a $N \times 2^N$ matrix where $r_{n,j} \in \{0, 1\}$ shows whether the nth trial results in a success (1) or not (0) in jth realization among 2^N possible sample paths. Let $O = \{o_{n,j}\}$ be a matrix with elements $o_{n,j}$ that shows the cumulative number of successes in the nth trial in jth sample path. That is,

$$o_{n,j} = \sum_{k=1}^{j} r_{k,j}, \qquad j = 1, \ldots, N, i = 1, \ldots, 2^N. \tag{13}$$

The probability of observing jth realization depends on the number of successes in that particular realization. Let $n_j = o_{N,j}$ be the number of successes among N trials in jth realization. Then

$$q_j = p^{n_j}(1-p)^{N-n_j}, \qquad j = 1, \ldots, 2^N. \tag{14}$$

Following these definitions, the following mathematical program gives the optimal state-independent synchronization policy:

$$\min_{\{\mathrm{h}_n\}} \quad \sum_{n=1}^{N} \left(c_{\mathrm{H}}\mathrm{h}_n + \sum_{j=1}^{2^N} q_j \left(EY_{n,j} - ES_{n,j}\right)\right)^2 \right) \tag{15}$$

$$EY_{n,j} = (N-n)p + o_{n,j}, \qquad n = 1,\ldots,N,\ j = 1,\ldots,2^N \tag{16}$$

$$ES_{n-1,j} - \mathrm{h}_n\mathrm{M} \le ES_{n,j} \le ES_{n-1,j} + \mathrm{h}_n\mathrm{M}, \qquad n = 1,\ldots,N,\ j = 1,\ldots,2^N \tag{17}$$

$$EY_{n,j} - (1-\mathrm{h}_n)\mathrm{M} \le ES_{n,j} \le EY_{n,j} + (1-\mathrm{h}_n)\mathrm{M}, \qquad n = 1,\ldots,N,\ j = 1,\ldots,2^N \tag{18}$$

$$q_j = p^{o_{N,j}}(1-p)^{N-o_{N,j}}, \qquad j = 1,\ldots,2^N, \tag{19}$$

$$ES_{0,j} = Np, \qquad j = 1,\ldots,2^N, \tag{20}$$

$$\mathrm{h}_n \in \{0,1\}, \qquad n = 1,\ldots,N. \tag{21}$$

In the above formulation, the objective function given in Equation (15) includes the calculation of the expectation considering all scenario realizations of $Y_n = j \in \{0,\ldots,n\}$, each scenario j with the corresponding probability q_j given in Equation (14). Equations (17) and (18) linearize Equation (10) where M is a big number. Note that while Equation (10) has the multiplication of the decision variables EY_n and ES_{n-1} with h_n, Equations (17) and (18) represent the same state transition function given in Equation (1) as linear inequalities.

3.2.2 Expected Value Approach

In this section, we use a formulation that does not take into account each possible scenario. On the contrary, at each period the prediction is made based on the expected state in which the system can be. This formulation is much simpler than the previous one given in Section 3.2.1, because it does not keep the consistency of the scenario in which the past prediction is correctly inherited from the previous period. On the contrary, in this expected value formulation the predictions from the previous period are averaged when synchronization does not occur, see Equations (24) and (25).

$$\min_{\{\mathrm{h}_n\}} \quad \sum_{n=1}^{N} \left(c_{\mathrm{H}}\mathrm{h}_n + \sum_{j=0}^{n} q_{n,j} \left(EY_{n,j} - ES_{n,j}\right)\right)^2 \right) \tag{22}$$

$$EY_{n,j} = (N-n)p + j, \qquad n = 1,\ldots,N,\ j = 0,\ldots,n \tag{23}$$

$$ES_{n,j} \ge \frac{pq_{n,j-1}ES_{n-1,j-1} + (1-p)q_{n,j}ES_{n-1,j}}{pq_{n,j-1} + (1-p)q_{n,j}} - \mathrm{h}_n\mathrm{M}, \qquad n = 1,\ldots,N,\ j = 0,\ldots,n \tag{24}$$

$$ES_{n,j} \le \frac{pq_{n,j-1}ES_{n-1,j-1} + (1-p)q_{n,j}ES_{n-1,j}}{pq_{n,j-1} + (1-p)q_{n,j}} + \mathrm{h}_n\mathrm{M}, \qquad n = 1,\ldots,N,\ j = 0,\ldots,n \tag{25}$$

$$EY_{n,j} - (1-\mathrm{h}_n)\mathrm{M} \le ES_{n,j} \le ES_{n,j} \le EY_{n,j} + (1-\mathrm{h}_n)\mathrm{M}, \qquad n = 1,\ldots,N,\ j = 0,\ldots,n \tag{26}$$

$$q_{n,j} = \binom{n}{j} p^j (1-p)^{n-j}, \qquad n = 1,\ldots,N,\ j = 0,\ldots,n \tag{27}$$

$$ES_{0,j} = Np, \qquad j = 1,\ldots,2^N, \tag{28}$$

$$\mathrm{h}_n \in \{0,1\},\ EY_{n,j} \ge 0,\ ES_{n,j} \ge 0 \qquad n = 1,\ldots,N\ j = 0,\ldots,n. \tag{29}$$

3.3 State-dependent Policy Formulation

For the state dependent case, the decision to synchronize depends on the number of trials and also the state of the system that is defined by the last observation. Let $\mathrm{h}_{n,k} \in \{0,1\}$ denote the synchronization decision at the nth trial when the number of successes observed is $Y_n = k$. The following formulation is used to determine the optimal state-dependent policy, $\mathbf{u}_n = \mathrm{h}_{\mathrm{n,y_n}}$, $n = 0, 1, \ldots, N$:

$$\min_{\{\mathrm{h}_{n,k}\}} \quad \sum_{n=1}^{N} \left(\sum_{k=0}^{n} c_{\mathrm{H}} \mathrm{h}_{n,k} + \mathbb{E}\left[(EY_n - ES_n))^2 \right] \right) \tag{30}$$

$$Y_n = y_n \qquad n = 1, \ldots, N \tag{31}$$

$$EY_n = (N-n)p + y_n, \qquad n = 1, \ldots, N \tag{32}$$

$$ES_n = ES_{n-1} + (EY_n - ES_{n-1})\mathrm{h}_{n,y_n}, \qquad n = 1, \ldots, N$$

$$ES_0 = Np \tag{33}$$

$$\mathrm{h}_{n,k} \in \{0, 1\}, \qquad n = 1, \ldots, N, \ k = 0, \ldots, n. \tag{34}$$

In the remainder of this section the scenario approach formulation is presented, the expected value formulation can easily be derived from this model and the one in Section 3.2.2.

Similar to the formulation for the state independent case, the formulation for the state dependent policy includes the realization of the random variable $Y_n = y_n$ given in Equation (31). As a result, the above formulation is also a stochastic nonlinear integer programming formulation. The solution for the general case can be determined by using a simulation optimization approach. For this specific case, a similar formulation as the one given for the state-independent case can be developed. The following Mixed Integer Quadratic Programming formulation uses all possible realizations of the sample path given in matrices R and O:

$$\min_{\{\mathrm{h}_{n,k}\}} \quad \sum_{n=1}^{N} \sum_{j=1}^{2^N} c_{\mathrm{H}} \mathrm{h}_{n,o_{n,j}} + q_j \left(EY_{n,j} - ES_{n,j} \right))^2 \tag{35}$$

$$EY_{n,j} = (N-n)p + o_{n,j}, \qquad n = 1, \ldots, N, \ j = 1, \ldots, 2^N \tag{36}$$

$$ES_{n-1,j} - \mathrm{h}_{n,o_{n,j}}\mathrm{M} \le ES_{n,j} \le ES_{n-1,j} + \mathrm{h}_{n,o_{n,j}}\mathrm{M}, \qquad n = 1, \ldots, N, \ j = 1, \ldots, 2^N \tag{37}$$

$$EY_{n,j} - (1 - \mathrm{h}_{n,o_{n,j}})\mathrm{M} \le ES_{n,j} \le EY_{n,j} + (1 - \mathrm{h}_{n,o_{n,j}})\mathrm{M}, \qquad n = 1, \ldots, N, \ j = 1, \ldots, 2^N \tag{38}$$

$$q_j = p^{o_{N,j}} (1-p)^{N - o_{N,j}}, \qquad j = 1, \ldots, 2^N, \tag{39}$$

$$ES_{0,j} = Np, \qquad j = 1, \ldots, 2^N \tag{40}$$

$$\mathrm{h}_{n,j} \in \{0, 1\}, \qquad n = 1, \ldots, N, \ j = 1, \ldots, 2^N. \tag{41}$$

4 Numerical Results

The solutions of the mathematical programs for the state-dependent and state-independent cases are determined by using CPLEX. The solutions are also verified by using the Genetic Algorithm available in Matlab that uses the expected cost function estimated using a simulation model coded in Matlab.

Figure 3 shows in red color the synchronization periods determined by the state-independent optimal policy for different c_{H} and p values using the scenario approach formulation. The synchronization action cost is scaled to show the relative accuracy improvement to be obtained with a synchronization action that requires a particular level of resource to be used. As expected, the number of synchronizations decreases with the increasing synchronization cost. The optimal periods for the synchronizations are around the equidistant points between 1 and N for a given number of synchronizations and depend on the values of c_{H} and p. Depending on c_{H} and p, there are multiple optimal solutions. For example, when only one synchronization is to be implemented, the optimal synchronization is around the 5th period, e.g either 4$^{\text{th}}$, 5$^{\text{th}}$ or 6$^{\text{th}}$ period. Similarly, if two synchronizations are determined to be implemented in 10 periods, these synchronizations are around the 3$^{\text{rd}}$and the 7$^{\text{th}}$ periods.

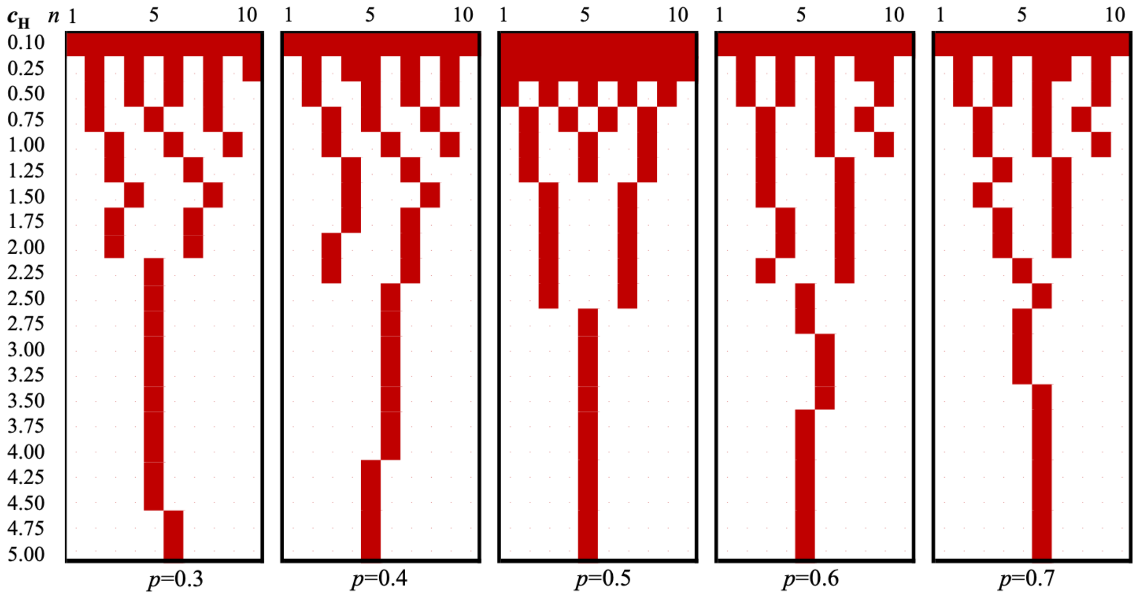

Figure 3: The synchronization periods determined by the state-independent optimal policy for different c_{H} and p values.

Figure 4 compares the synchronization periods determined by the state-dependent and the state-independent policy for $c_H = 2$ and $p = 0.5$. While the state-independent policy sets two synchronization points in periods 3 and 7, the state-dependent policy sets a higher number of synchronization points. However, based on the likelihood of observing certain states, the average number of synchronizations is fewer than the state-independent case. As a result, the average total cost obtained by the state-dependent policy is lower than the cost of the state-independent policy.

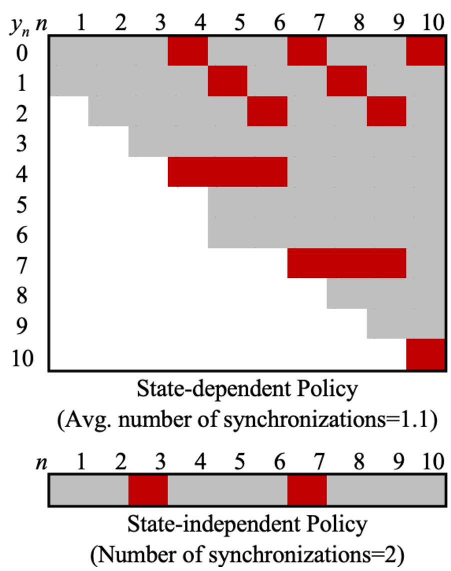

Figure 4: Comparison of the synchronization periods determined by the state-dependent and the state-independent optimal policy ($c_{\mathrm{H}} = 2$ and $p = 0.5$).

Figure 5 and 6 show for $p = 0.5$ the average total cost and the average number of synchronizations or different values of synchronization cost c_H respectively. The state-independent and state-dependent solutions as well as the solution obtained under the perfect information case for the state-independent policy are depicted. As expected, the state-dependent policy yields a lower average total number of synchronizations and therefore a lower average total cost. The benefit of using a state-dependent synchronization policy increases as the synchronization cost increases and gets closer to the solution under the perfect information setting.

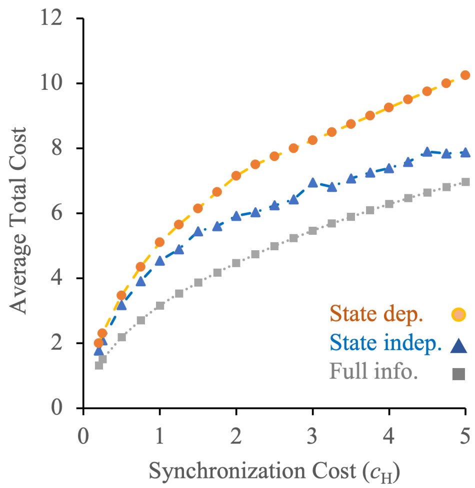

Figure 5: Average total cost vs c_H

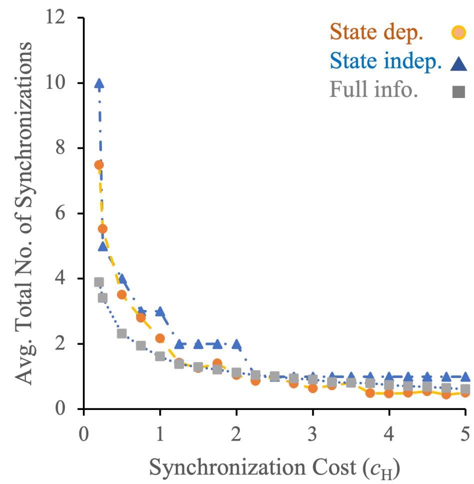

Figure 6: Average number of synchronizations vs c_H.

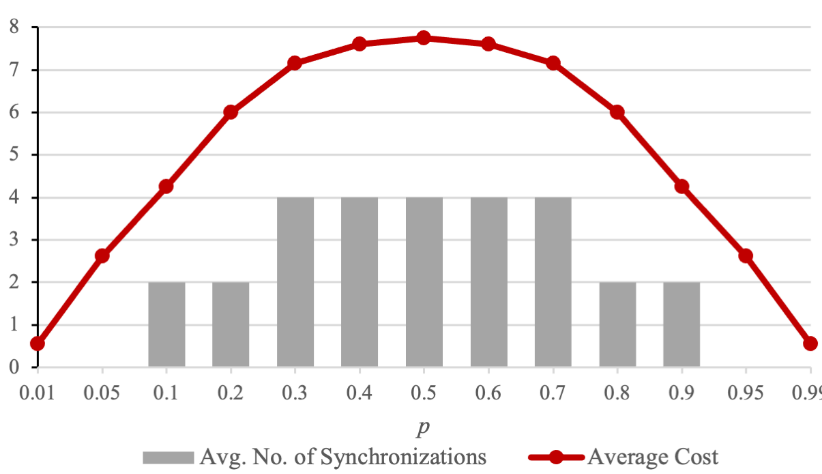

Figure 7: Average total cost and number of synchronizations vs p for $c_H = 2$ (state–independent policy with scenario approach formulation).

Figure 7 shows the average total cost vs p for $c_H = 2$ for the state–independent case. When $p = 0.5$ the system is affected by the highest variability and therefore the total cost reaches the peak as well as the number of synchronizations (2 out of 10 periods). As p departs from 0.5, the variability decreases; as a result, it is easier to make predictions and a lower number of synchronizations is needed.

5 CONCLUSIONS

In this paper, we define and analyze the problem of determining the optimal synchronization periods dynamically for a digital twin that tracks the performance measure of a real system. An optimal control problem is introduced to determine when the simulation should be synchronized with the real system in order to minimize the total costs related to inaccuracy of the performance measures obtained by simulation and the costs related to using the resources to improve the accuracy of the predictions obtained by the digital twin. An optimal simulation synchronization policy determines whether to synchronize the simulation dynamically at each decision epoch based on the data collected from the system. We present a stylistic model for a special case, formulate the optimal synchronization problem as a mathematical program and determine the state-dependent and state-independent optimal policies.

The analysis of the specific case shows that the optimal policy is not synchronizing the system at all decision epochs. When a state-independent policy is used, the synchronization periods are equally spaced along the planning horizon. A state-dependent policy defines a higher number of synchronization periods depending on the observations and the trial number but the average number of synchronization is lower than the state-independent case. As system variability increases, synchronizing becomes more important.

This work can be extended in several ways. The solution of the infinite horizon case, characterizing the optimal policies for the finite and infinite horizon cases, and defining, formulating, analyzing the optimal detail level and parameter update problems, and the analysis of the problem with reinforcement learning are left for future research. is left for future research.

REFERENCES

Barricelli, B. R., E. Casiraghi, and D. Fogli. 2019. "A survey on digital twin: Definitions, characteristics, applications, and design implications". *IEEE access* 7:167653–167671.

Cardin, O., and P. Castagna. 2011. "Proactive production activity control by online simulation". *International Journal of Simulation and Process Modelling* 6(3):177–186.

Grandview Research 2021. "Simulation Software Market Size, 2021-2028". Technical report.

Hanisch, A., J. Tolujew, and T. Schulze. 2005. "Initialization of online simulation models". In *Proceedings of the 2005 Winter Simulation Conference*, edited by M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 1795–1803. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kritzinger, W., M. Karner, G. Traar, J. Henjes, and W. Sihn. 2018. "Digital Twin in manufacturing: A categorical literature review and classification". *IFAC-PapersOnLine* 51(11):1016–1022.

Powell, W. B. 2021. "From reinforcement learning to optimal control: A unified framework for sequential decisions". In *Handbook of Reinforcement Learning and Control*, 29–74. Springer.

Sargent, R. G. 2010. "Verification and validation of simulation models". In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Ycesan, 166–183. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Tao, F., Q. Qi, L. Wang, and A. Nee. 2019. "Digital twins and cyber–physical systems toward smart manufacturing and industry 4.0: Correlation and comparison". *Engineering* 5(4):653–661.

Uhlemann, T. H.-J., C. Lehmann, and R. Steinhilper. 2017. "The digital twin: Realizing the cyber-physical production system for industry 4.0". *Procedia CIRP* 61:335–340.

AUTHOR BIOGRAPHIES

Barış Tan is a Professor of Operations Management and Industrial Engineering at Koç University. His research interests include design and control of manufacturing systems, supply chain management, and stochastic modelling. His e-mail address is btan@ku.edu.tr. His website is http://home.ku.edu.tr/∼btan/.

Andrea Matta is Professor of Manufacturing at Politecnico di Milano, where he currently teaches integrated manufacturing systems and manufacturing. His research area includes analysis, design and management of manufacturing and health care systems. He is Editor in Chief of the Flexible Services and Manufacturing Journal. His email address is andrea.matta@polimi.it.