scientific reports

OPEN

Synchronizing controlled logistics terminals between simulated and visualized production lines using an ASTAK method

Zixiao Wang[1]✉, Yue Dong[1], Shengguo Wang[2] & Xinxiang Zhang[3]

In a fully automated factory, the Visualized Production Line serves as a crucial tool for assisting personnel to monitor and manage the manufacturing process. The synchronization between the visualized line and the actual production line significantly impacts the efficiency of production supervision. This article proposes a method for controlling the logistics terminals, which encompasses three steps: animation simplification, timing alignment, and keyframe synchronization (hereinafter referred to as ASTAK). This method aims to achieve precise synchronization between the Simulated Production Line and the Visualized Production Line when the process data of the simulated line is not directly accessed. Then, the experiments demonstrate that the proposed method reduces the time difference between the simulated and visualized production lines to an average of 0.08 s with a synchronization rate of 99.97%, which further verifies the effectiveness and superiority of the proposed method over some other state-of-the-art methods.

Keywords Synchronization, Controllable logistics terminal, Animation simplification, Timing alignment, Keyframe

In recent years, various unmanned production lines, including both process-oriented and discrete manufacturing processes, can effectively fulfill production tasks within their respective processes[1]. Since the establishment of these production lines, the issue of process supervision has consistently been a key concern. In this context, the emergence of visualized production lines (also known as Digital Factories or Digital Twin Models in some contexts) based on digital twin technology has been widely applied[2–9]. These visualized production lines are designed to present the production process, highlight process parameters, and allow for the tracking of production progress. However, the development of unmanned factories is still premature, resulting in inaccurate synchronization between actual production lines and visualized production lines[10]. In this article, an ASTAK method is proposed to address the synchronization challenges between simulated and visualized production lines. This method integrates three core innovations: (1) Animation Simplification: pre-simplified animations that replace real-time 3D modeling to optimize computational efficiency and synchronization between virtual and actual production lines, (2) Timing Alignment: stage-based process division with synchronized timing that ensures the visualized production line mirrors the actual production flow, and (3) Keyframe Synchronization: aligning equipment actions with animation keyframes that achieves movement consistency and enhances visual synchronization accuracy.

Research background and related studies

The technology of utilizing visualized production lines to present the actual production process is known as digital twin. The digital twin has been attempted and has achieved certain success in engineering construction, and intelligent manufacturing fields in the last decade. The digital twin is a technology that utilizes digital methods to simulate physical entities. In the past, a general digital twin was achieved based on multiple sensors and digital 3D models[11–13]. Engineers needed to build digital 3D models in a virtual three-dimensional space according to physical entities and rely on the sensors in the real world to collect operational data from the physical entities. Based on the operational data, real-time calculations were made to estimate the movement

[1]School of Information and Communication Engineering, Communication University of China, Beijing 100024, China. [2]China Ordnance Industry Survey and Geotechnical Institute Co., Ltd, Beijing 100053, China. [3]Department of Electrical and Computer Engineering, Southern Methodist University, Dallas, TX 75205, USA. ✉email: wangzixiao@cuc.edu.cn

of the 3D models in the virtual three-dimensional space. Most traditional digital twins are realized based on real-time calculations of 3D models[14,15]. In digital factories, a system that digitizes production lines using digital twin methods is referred to as visualized production lines. In this article, a physical production line simulating an actual production line is referred to as a simulated production line. By applying digital twin technology to the simulated production line, a matching visualized production line can be built. In order to make the visualized production line play a better role in monitoring production, it is necessary to improve the synchronization rate between the visualized production line and the simulated production line.

There are some previous studies on the synchronization between visualized production lines and actual production lines[16,17]. Holger Zipper transformed the problem of synchronizing the status of the actual production line and the visualized production line into an optimization problem, in which the time difference between the actual production line output and the visualized production line output was taken as the optimization objective[18,19]. However, this method requires real-time processing of data from equipment in the production line, which poses a certain security risk to the actual production line. To this end, Sangmi Park and Changhee Hong proposed a real-digital synchronization method for inspecting underground integrated pipe racks based on a single camera implementation[20]. In this reference, by introducing deep learning technology, the coordinate system can be transformed and projected, enabling the mapping of real-world pipe racks onto the visualized production line. However, in the real production process, the reliability of cameras often cannot meet the requirements of all scenarios. To solve this problem, Adrian Kampa imported real-time production data to a database and synchronized the visualized production line with the actual production line by maintaining synchronization between the Work-in-Process states and the database[21]. However, there remains considerable potential to enhance the synchronization effect attained via the Work-in-Process status.

In this article, the Controlled Logistics Terminals (hereinafter referred to as CLT) in a simulated production line of an unmanned factory are used as an example for the experiment with the ASTAK method. Through the three steps of animation simplification, timing alignment, and keyframe synchronization, the consistency in the movements of the CLT between these two production lines has been achieved[22–28].

The contributions to this paper are listed as follows:

- Instead of real-time calculation of 3D modeling, the utilization of animation is proposed to present equipment on the visualized production line. For equipping with adjustable motion paths, the simplification of animated resources for the visualized production line is introduced into the synchronization method. These approaches significantly enhance the computational efficiency and synchronization effect, which provides a novel perspective on the synchronization issue between the visualized production line and the actual production line.
- The production process of equipment is divided into multiple stages, and introducing timing alignment is proposed as a crucial component of the synchronization algorithm for both production lines. This approach ensures that the visualized production line precisely mirrors the actual production flow.
- The alignment of key actions of equipment with keyframes of the animated resources for the visualized production line is proposed to achieve consistency in the movements of equipment across both production lines. This significantly enhances the visual experience on front-end devices for the visualized production line, offering an optimized solution for precise synchronization between the visualized and actual production lines.

The structure of this article is as follows: Sect. 1 introduces the background of this study, the equipment and process flow involved in the research. Section 2 presents the methods involved in the research. Section 3 conducts experiments based on the proposed method and obtains experimental results. Section 4 discusses potential issues that may still exist in the research. Section 5 presents the final conclusion of the proposed method.

Main components of the production line

In this section, the main components of the production line are described. This paper takes a simulated production line from a specific unmanned factory and its accompanying visualized production line as the subjects of research. Furthermore, the CLT is applied to research the synchronization of the two production lines.

Controlled logistics terminals

CLT is a key component in the simulated production line for transporting materials. During the process of the simulated production line, CLT transports materials C and D to the lifting platform. The lifting platform and the six-axis manipulator transport the materials to the reaction kettle[29]. CLT then returns to the charging pile to wait. Its operation flowchart is shown in Fig. 1.

The operating speed of CLT can be adjusted based on production needs. When production tasks are urgent, CLT will increase its operating speed, and vice versa.

Figure 2illustrates the controlled process of CLT within the system. The system manager can choose the manual mode to directly send orders to manage the movements of the CLT[30]. Alternatively, they can select the automatic mode that allows the pre-programmed PLC to fully handle the instruction transmission for CLT[31–33]. The PLC communicates with CLT through a gateway, enabling CLT to complete the designated actions. CLT then sends feedback to the PLC via the gateway again.

Simulated production line and visualized production line

The simulated production line refers to a physical production line designed to replicate the actual manufacturing processes of a factory. It allows for the assessment of the effectiveness and reliability of various equipment, systems, and programs. The visualized production line is a software system that visually processes the simulated production line and displays the production process. The visualized production line can dynamically simulate

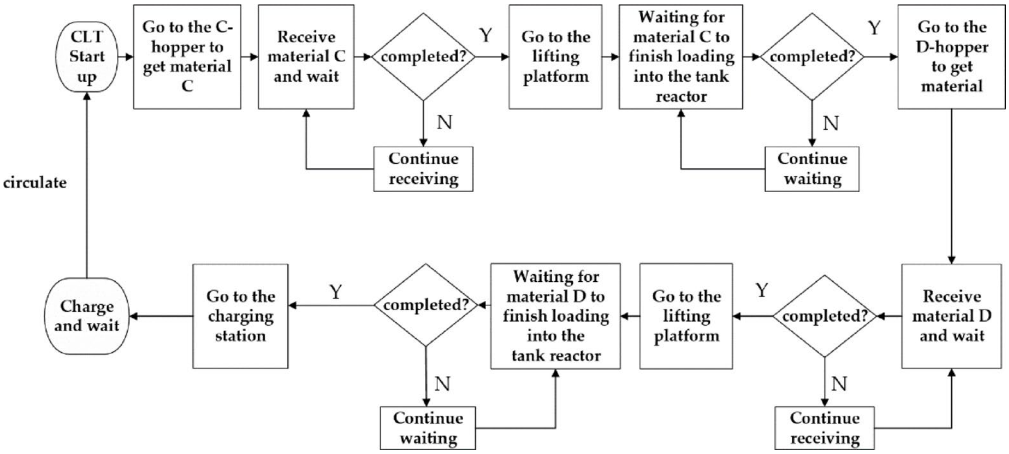

CLT Start up
Go to the C-hopper to get material C
Receive material C and wait
completed?
Y
N
Continue receiving
Go to the lifting platform
Waiting for material C to finish loading into the tank reactor
completed?
Y
N
Continue waiting
Go to the D-hopper to get material
circulate
Receive material D and wait
completed?
Y
N
Continue receiving
Go to the lifting platform
Waiting for material D to finish loading into the tank reactor
completed?
Y
N
Continue waiting
Go to the charging station
Charge and wait

Fig. 1. CLT operation flowchart.

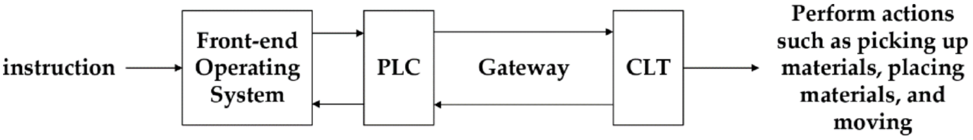

instruction
Front-end Operating System
PLC
Gateway
CLT
Perform actions such as picking up materials, placing materials, and moving

Fig. 2. CLT control flowchart.

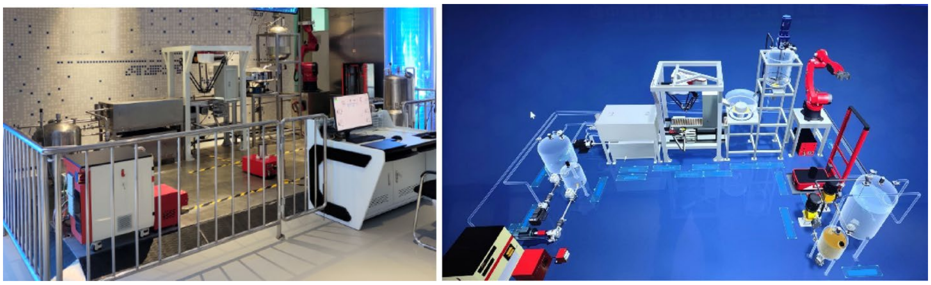


Fig. 3. Simulated production line (left) and visualized production line (right). Note that all the equipment inside the metal railing in the left picture is simulated production line equipment. The tables and computers outside the railing are used by operators to manage the production line.

the entire production line based on the requirements of the production process and the actual status of the on-site equipment. It displays status information regarding logistics, information flow, and control flow within the system[34]. The simulated production line and the visualized production line are shown in Fig. 3.

Process flow

The simulated production line is designed to mix and separate liquid components A and B, as well as solid materials C and D. As shown in Fig. 4, liquid phases A and B are dosed into the reactor by metering pumps. Solid materials C and D are picked up by the CLT once discharged through the hopper. These materials are then transported to the lifting platform. After ascending to the second level of the dosage position, the six-axis manipulator adds them to the reactor. Once the reactor mixes A, B, C, and D evenly, the mixed materials are discharged through the discharge valve into the water sifter. The liquid-phase components, after the solid materials are sieved out, flow into the curvature. A and B are separated and directed into the recycling tank by adjusting the setting. The solid material screened out by the water sifter travels along the conveyor belt at a uniform speed. Finally, material D is detected by an intelligent camera and then picked up by the spider manipulator for placement into the raw material D hopper[35]. The solid material C enters the raw material C hopper with the conveyor belt.

When encountering a special scenario, the visualized production line does not rely on calculating the motion paths of 3D models for display but instead creates a large number of corresponding animations for fixed equipment movements. Playing fixed animations only involves retrieving animation resources and does not involve the

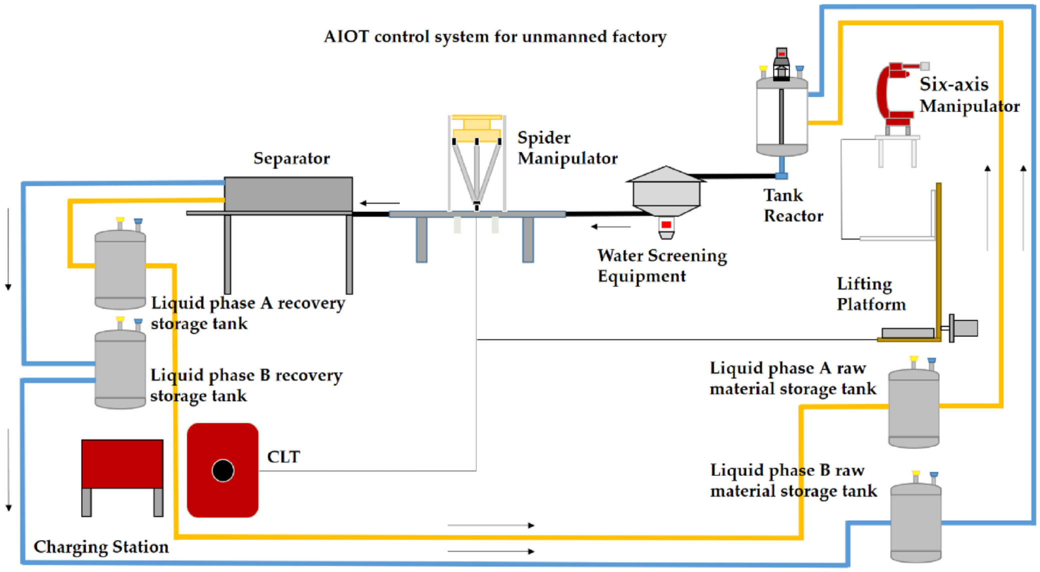

AIOT control system for unmanned factory
Six-axis Manipulator
Spider Manipulator
Separator
Tank Reactor
Water Screening Equipment
Lifting Platform
Liquid phase A recovery storage tank
Liquid phase B recovery storage tank
Liquid phase A raw material storage tank
Liquid phase B raw material storage tank
CLT
Charging Station

Fig. 4. The processing flowchart. Note that the blue pipeline represents the delivery of aqueous liquid component A, the yellow pipeline represents the delivery of oily liquid B, and the black pipeline represents the delivery of the mixed liquid. The other colors in the figure represent the actual appearance of the equipment in reality.

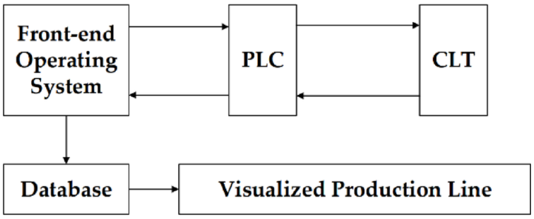


Fig. 5. The role of database.

calculation of 3D modeling motion paths or rendering of images. By combining modeling with animation, the workload on the computer is reduced to a certain extent, improving the efficiency of synchronization.

Methodologies

To coordinate the simulated production and the visualized production line, a line synchronization has been developed. In this section, the ASTAK production line synchronization method has been described. The ASTAK method includes three main steps: animation simplification, timing alignment, and keyframe synchronization, which are described in the following three sections, respectively.

In the production process, the three steps of the ASTAK method act on different objects. The animation simplification step acts on the animations played by the visualized production line itself, which are completed before the production line starts running. During the production process, it is the back-end system of the visualized production line that determines which animation resource should be played at each moment after loading the data from the database. The timing alignment step affects the back-end of the visualized production line. The algorithm described in this step then determines the duration of the playback of this animation resource. Keyframe synchronization primarily acts on the front-end system of the visualized production line. During the playback of a certain animation, the method described in this step adjusts the frame rate of the animation locally to achieve synchronization between the visualized production line and the simulated production line. The notations used in this paper are shown in Table 1.

Animation simplification

Within the framework of visualized production lines, the plethora of embedded animation resources often translates into extended retrieval and loading durations. In this case, animation simplification is a way to improve efficiency. The animation assets utilized in visualized production lines are tailored according to the motion paths of the equipment being simulated. For devices with variable paths, if there is overlap in their movement trajectories, using the same animation resources for the sections of shared paths will not compromise the visual effect of the animation. Therefore, by optimizing the paths, simplification of animation resources is effectively attained.

The path optimization approach discussed in this paper is applicable to equipment that performs both horizontal and vertical movements, such as devices guided by horizontally or vertically arranged magnetic strips (CLT). The animation simplification method is specifically discussed as follows:

The process of moving a device from one point to another is considered a task. In a production cycle, the device has a total of n tasks, resulting in a task sequence of $\{t_1, t_2, \ldots, t_n\}$. Each task corresponds to a path, namely $\{\boldsymbol{\rho_1}, \boldsymbol{\rho_2}, \ldots, \boldsymbol{\rho_n}\}$. The starting and ending points of $\boldsymbol{\rho_i}$ are determined by t_i, and the task sequence determines the order of movement of each device between various points. In actual production, the coordinates of the points passed by the device are fixed. A path $\boldsymbol{\rho_i}$ composed of γ points $\{(x_{i,1}, y_{i,1}), (x_{i,2}, y_{i,2}), \ldots, (x_{i,\gamma}, y_{i,\gamma})\}$ is randomly generated between the starting and ending points of a task. $\boldsymbol{\rho_i}$ satisfies Eq. (1):

$$d(\boldsymbol{\rho}_{\boldsymbol{i}}) = \sum\nolimits_{j=1}^{\gamma-1} \max(|x_{i,j+1} - x_{ij}|, |y_{i,j+1} - y_{i,j}|) \tag{1}$$

where γ is determined by the required precision under specific production tasks, and $d(\boldsymbol{\rho_i})$ represents the Manhattan distance between the starting and ending nodes of path $\boldsymbol{\rho_i}$. After generating multiple paths, a comparison of overlap degrees needs to be conducted among the paths to select the one that requires the fewest animation resources. To compare the overlap degrees, each path needs to be divided into a series of line segments $\{s_{i,1}, s_{i,2}, \ldots, s_{i,a_i}\}$ and corners $\{\theta_{i,1}, \theta_{i,2}, \ldots, \theta_{i,b_i}\}$. Path $\boldsymbol{\rho_i}$ has a total of a_i line segments and b_i turning angles, with the absolute value of all corners being 90° or 180°. Let $D_{i,k}$ represent the direction of the k-th line segment in $\boldsymbol{\rho_i}$, as shown in (2):

$$D_{i,k} = \begin{bmatrix} \cos\theta_{i,k} & -\sin\theta_{i,k} \\ \sin\theta_{i,k} & \cos\theta_{i,k} \end{bmatrix} D_{i,k-1} \tag{2}$$

Let the starting point of the line segment $s_{i,j}$ be $(x_{i,j}^s, y_{i,j}^s)$ and the ending point be $(x_{i,j}^e, y_{i,j}^e)$. If the path $\boldsymbol{\rho_{i-1}}$ of the previous task has a turn after traversing all the line segments in the path, then let $D_{i,0}$ equal to $D_{i-1,bi}$; otherwise, let $D_{i,0}$ equal to 1. The path $\boldsymbol{\rho_i}$ can then be expressed as in Eq. (5):

$$\boldsymbol{\rho}_{\boldsymbol{i}} = \{(x_{i,1}^s, y_{i,1}^s), d(s_{i,1}) D_{i,0}, d(s_{i,2}) D_{i,1}, \ldots, d(s_{i,a_i}) D_{i,a_i-1}\} \tag{3}$$

$$O(s_{i,j}, s_{m,n}) = \begin{cases} min(y_{(i,j)}^e, y_{(m,n)}^e) - max(y_{(i,j)}^s, y_{(m,n)}^s) & \begin{array}{c} x_{(i,j)}^s = x_{(i,j)}^e = x_{(m,n)}^s = x_{(m,n)}^e| \\ |min(y_{(i,j)}^e, y_{(m,n)}^e) > max(y_{(i,j)}^s, y_{(m,n)}^s) \end{array} \\ min(x_{(i,j)}^e, x_{(m,n)}^e) - max(x_{(i,j)}^s, x_{(m,n)}^s) & \begin{array}{c} y_{(i,j)}^s = y_{(i,j)}^e = y_{(m,n)}^s = y_{(m,n)}^e \\ ||min(x_{(i,j)}^e, x_{(m,n)}^e) > max(x_{(i,j)}^s, x_{(m,n)}^s) \end{array} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Using this function, one can define a function $S(\boldsymbol{\rho_i}, \boldsymbol{\rho_j})$ to calculate the overlap degree between two paths $\boldsymbol{\rho_i}$ and $\boldsymbol{\rho_j}$, as shown in (5):

$$S(\boldsymbol{\rho}_{\boldsymbol{i}}, \boldsymbol{\rho}_{\boldsymbol{m}}) = \sum\nolimits_{j=1}^{a_i} \sum\nolimits_{n=1}^{a_m} O(s_{i,j}, s_{m,n}) \tag{5}$$

During the process of generating the path with the highest overlap degree, multiple sets of solutions need to be generated for the complete path $\boldsymbol{\rho} = \{\boldsymbol{\rho_1}, \boldsymbol{\rho_2}, \ldots, \boldsymbol{\rho_n}\}$, and their overlap degrees are calculated accordingly. The path with the highest overlap degree $\boldsymbol{\rho}*$ is finally chosen as the final solution. This process is illustrated in (6):

$$\boldsymbol{\rho}* = \underset{\boldsymbol{\rho}}{\text{argmax}} \sum\nolimits_{i=2}^{n} \sum\nolimits_{j=1}^{i-1} S(\boldsymbol{\rho}_{\boldsymbol{i}}, \boldsymbol{\rho}_{\boldsymbol{j}}) \tag{6}$$

The main steps of the animation simplification method are given in Algorithm 1. Note that η represents the minimum number of iterations required to generate the path with the highest overlap degree under the constraints of a specific production environment, and other notations are defined in Table 1.

The operation of CLT applied in this paper involves four nodes and five tasks, as shown in Fig. 6. These four nodes, respectively, represent the charging station, material C hopper, material D hopper, and lifting platform.

A coordinate system is established on the plane where CLT operates. By continuously selecting suitable paths $\boldsymbol{\rho_1}$ for task t_1 and equations corresponding to overall paths $\boldsymbol{\rho}$, different path scenarios are obtained, as shown in Fig. 8. By calculating the path overlap, the performance of different scenarios can be compared. The above two examples are shown in Fig. 7. The overlap score of scenario 2 is higher than that of scenario 1. In scenario 2, only two animation resources, from node p_1 to p_3 and from node p_4 to node o, are needed to visualize the five tasks. An animation resource presents different scenes of CLT for different tasks by playing partial clips and reversing them. By simplifying the animation resources and streamlining the animation resources, we have achieved a noticeable enhancement in the synchronization effect between the visual and simulated production lines. This improvement will be substantiated through the experiments outlined in this paper. Note that scenario 2 is also selected as the path for CLT operation in the follow-up case study.

Timing alignment

Figure 8 is a schematic diagram of various phases of the CLT task. Using the algorithm provided in the Animation Simplification section, the path optimization for tasks t_1 to t_5 in the figure is completed, achieving

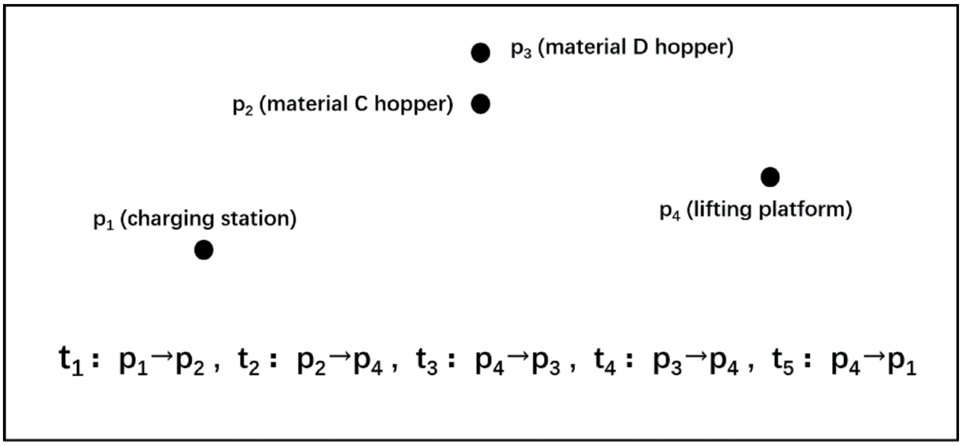

p_3 (material D hopper)
p_2 (material C hopper)
p_1 (charging station)
p_4 (lifting platform)
t_1 : $p_1 \rightarrow p_2$, t_2 : $p_2 \rightarrow p_4$, t_3 : $p_4 \rightarrow p_3$, t_4 : $p_3 \rightarrow p_4$, t_5 : $p_4 \rightarrow p_1$

Fig. 6. The nodes and tasks involved in the operation of CLT.

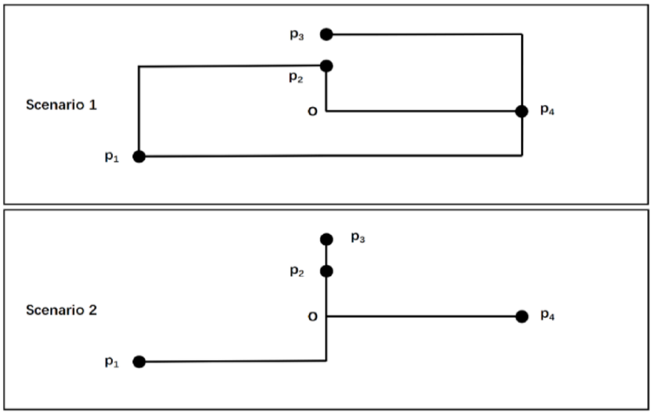


Fig. 7. Two path scenarios.

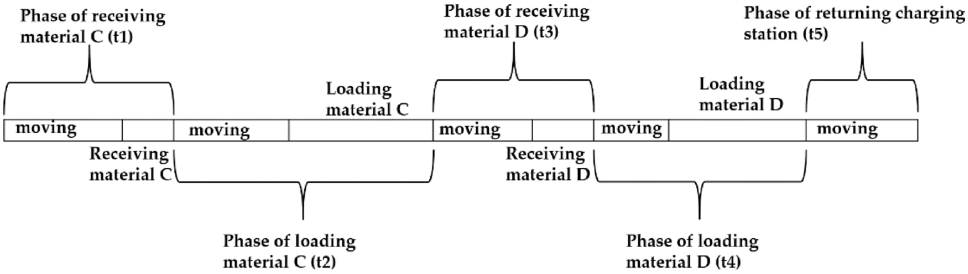

Phase of receiving material C (t1)
Phase of receiving material D (t3)
Phase of returning charging station (t5)
Loading material C
Loading material D
moving
moving
moving
moving
moving
Receiving material C
Receiving material D
Phase of loading material C (t2)
Phase of loading material D (t4)

Fig. 8. Schematic diagram of each phase of CLT task.

the simplification of animation resources. This article develops a time alignment method to further enhance the synchronization effect of the visual production line in response to changes in the production situation of the simulated production line. During the synchronization process between the visual production line and the actual production line, the desynchronization of equipment operation screens often occurs with changes in production conditions. By introducing time alignment, the working cycle of the equipment in the entire production line is divided into multiple stages. The back-end system of the visual production line continuously loads the time data of the start and end of each task stage and the equipment operating speed data from the database. The back-end system also determines the playback duration of the animation and adjusts the animation playback speed. This ensures the synchronization of the screen when changes in production situations occur.

$$\epsilon_i = \epsilon\,_i + \mu\,_i \tag{7}$$

The time ϵ_i required for each task t_i includes both the moving and stationary parts of the equipment, as shown in Eq. (7):

Since the visualized production line directly loads the equipment's movement speed information from the database, ε_i is accurately calculated on the back-end system of the visualized production line, as shown in Eq. (8). Note that $d(s_{i,j})$ represents the length of the line segment $s_{i,j}$.

$$\epsilon_{\ i} = \sum_{j=1}^{a_i} d\left(s_{i,j}\right) v_{ij} + \sum_{k=1}^{b_i} \theta_{\ ik}\omega_{\ ik} \tag{8}$$

Since μ_i is affected by real-time, actual production conditions are hard to determine. However, the time when the equipment remains stationary during a task phase can be estimated based on the equipment's movement time $\lambda_{\ i}$ and stationary time $\delta_{\ i}$ in the same phase during the previous operation cycle. In usual actual production, these two times often show a positive correlation. The estimation method is shown in Eq. (9):

$$\mu_{\ i} = \frac{\epsilon_{\ i}}{\lambda_{\ i}}\delta_{\ i} \tag{9}$$

By using Eqs. (7), (8), and (9), an estimated ϵ_i value can be obtained, which allows us to determine the playback speed multiplier for the animation corresponding to the i-th task phase, as shown in Eq. (10). The front-end system of the visualized production line adjusts the frame rate to play the animation of a specific task phase at a speed multiplier of κ.

$$\kappa_{\ i} = \frac{\partial_i}{\epsilon_i} \tag{10}$$

Obtaining the playback speed multiplier for each task phase animation is only part of achieving time alignment, and determining the start and stop times for each phase is also necessary. When the back-end system of the visualized production line loads r_i from the database, it begins to play the animation of the i-th task phase. However, since the calculated ϵ_i is not an exact value, the time point at which a certain phase animation finishes playing may be earlier or later than $\varphi_{\ i}$. Note that $\varphi_{\ i}$ can only be loaded by the visualized production line back-end system at the actual end of the i-th task phase and is unknown during animation playback.

Therefore, different scenarios need to be further discussed. The actual ϵ_i is obtained by subtracting $\varphi_{\ i}$ and r_i. If the animation has not finished playing yet but $\varphi_{\ i}$ has already been loaded, the back-end system of the visualized production line will directly stop playing the animation for phase i, and α_i is set to 2. If $\varphi_{\ i}$ has not been loaded by the time the animation is played, the animation for phase i on the visualized production line will keep playing the last frame after playback until $\varphi_{\ i}$ is loaded, and α_i is set to 0. If the animation has just finished playing when $\varphi_{\ i}$ is loaded, no further processing is required, and α_i is set to 1. Since the stationary state of the equipment is generally selected as the second half of a task phase during phase division, it will not cause a noticeable lack of fluidity in the animation. Figure 9 shows that the solution for the $\varphi_{\ i}$ has not been loaded by the time the animation is played.

Figure 9. The solution for the $\varphi_{\ i}$ has not been read by the time the animation is played.

The main steps in the timing alignment method are given in Algorithm 2. Note that the notions are defined in Table 1.

Keyframe synchronization

The playback duration of the animation for each stage of the task is synchronized with the actual situation by using the time alignment method, but there may still be individual actions that are not synchronized. By treating some actions as keyframes in the animation, visual synchronization between the simulated production line and the visual production line is achieved.

During the operation process, the PLC records and stores data at operating nodes in the database. The animation frames correspond to the equipment actions at these operating nodes, which are regarded as keyframes. When the playback moment of a keyframe deviates from the recorded moment of an operating node by more than 0.2 s, the human eye will perceive a significant desynchronization. When the n-th operating node is reached, the database records this moment as τ_n, and the back-end system of the visual production line will compare it with ψ_n. The β_n is used to indicate whether the animation is accelerated or decelerated at the n-th keyframe. If the difference between τ_n and ψ_n is within 0.2 s, it means that the visualized production line is synchronized, and β_n is set to 0. If τ_n is more than 0.2 s ahead of ψ_n, it means that the animation lags behind

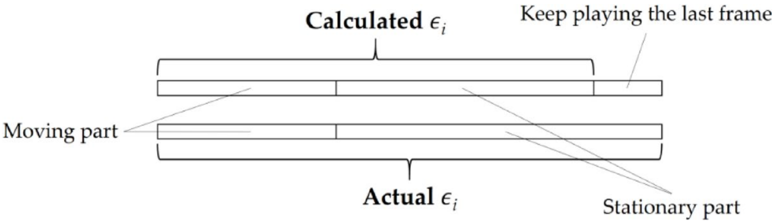
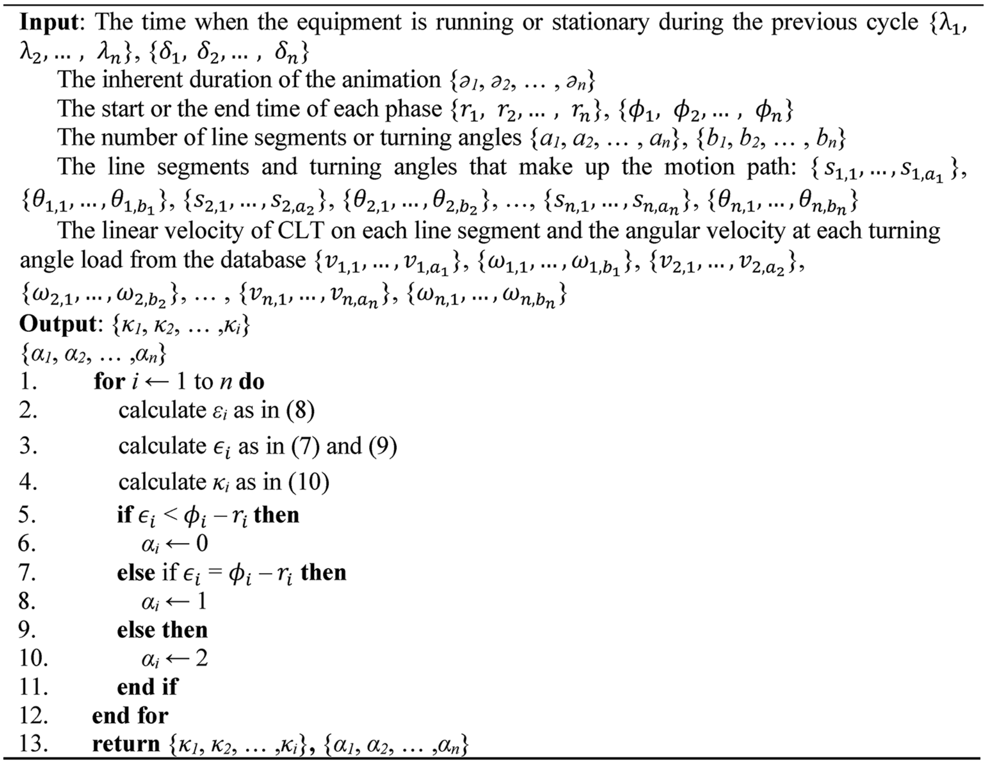
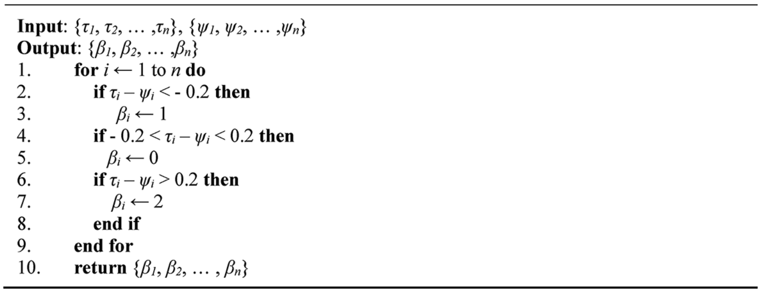


Fig. 9. The solution for the ϕ_i has not been read by the time the animation is played.



```
Input: The time when the equipment is running or stationary during the previous cycle $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$, $\{\delta_1, \delta_2, \dots, \delta_n\}$
    The inherent duration of the animation $\{\partial_1, \partial_2, \dots, \partial_n\}$
    The start or the end time of each phase $\{r_1, r_2, \dots, r_n\}$, $\{\phi_1, \phi_2, \dots, \phi_n\}$
    The number of line segments or turning angles $\{a_1, a_2, \dots, a_n\}$, $\{b_1, b_2, \dots, b_n\}$
    The line segments and turning angles that make up the motion path: $\{s_{1,1}, \dots, s_{1,a_1}\}$, $\{\theta_{1,1}, \dots, \theta_{1,b_1}\}$, $\{s_{2,1}, \dots, s_{2,a_2}\}$, $\{\theta_{2,1}, \dots, \theta_{2,b_2}\}$, …, $\{s_{n,1}, \dots, s_{n,a_n}\}$, $\{\theta_{n,1}, \dots, \theta_{n,b_n}\}$
    The linear velocity of CLT on each line segment and the angular velocity at each turning angle load from the database $\{v_{1,1}, \dots, v_{1,a_1}\}$, $\{\omega_{1,1}, \dots, \omega_{1,b_1}\}$, $\{v_{2,1}, \dots, v_{2,a_2}\}$, $\{\omega_{2,1}, \dots, \omega_{2,b_2}\}$, … , $\{v_{n,1}, \dots, v_{n,a_n}\}$, $\{\omega_{n,1}, \dots, \omega_{n,b_n}\}$
Output: $\{\kappa_1, \kappa_2, \dots, \kappa_i\}$
$\{\alpha_1, \alpha_2, \dots, \alpha_n\}$
for $i \leftarrow 1$ to $n$ do
    calculate $\varepsilon_i$ as in (8)
    calculate $\epsilon_i$ as in (7) and (9)
    calculate $\kappa_i$ as in (10)
    if $\epsilon_i < \phi_i - r_i$ then
        $\alpha_i \leftarrow 0$
    else if $\epsilon_i = \phi_i - r_i$ then
        $\alpha_i \leftarrow 1$
    else then
        $\alpha_i \leftarrow 2$
    end if
end for
return $\{\kappa_1, \kappa_2, \dots, \kappa_i\}$, $\{\alpha_1, \alpha_2, \dots, \alpha_n\}$
```

Algorithm 2. Timing alignment method.



```
Input: $\{\tau_1, \tau_2, \dots, \tau_n\}$, $\{\psi_1, \psi_2, \dots, \psi_n\}$
Output: $\{\beta_1, \beta_2, \dots, \beta_n\}$
for $i \leftarrow 1$ to $n$ do
    if $\tau_i - \psi_i < -0.2$ then
        $\beta_i \leftarrow 1$
    if $-0.2 < \tau_i - \psi_i < 0.2$ then
        $\beta_i \leftarrow 0$
    if $\tau_i - \psi_i > 0.2$ then
        $\beta_i \leftarrow 2$
    end if
end for
return $\{\beta_1, \beta_2, \dots, \beta_n\}$
```

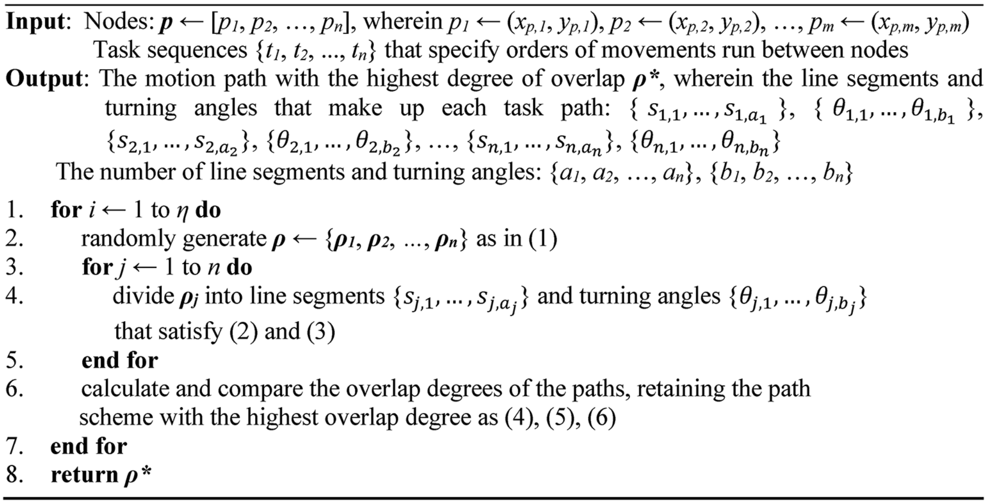
Algorithm 3. Keyframe Synchronization Method.

the actions of the equipment in the simulated production line at the n-th keyframe. So β_n is set to 1, and the animation will double the play speed within one second. Conversely, if τ_n lags behind ψ_n by more than 0.2 s, it means that the animation is ahead of the actions of the equipment in the simulated production line at the n-th keyframe. So β_n is set to 2, and the animation will reduce the play speed by a half within one second. The main steps in the keyframe synchronization method are given in algorithm 3.

As seen from Table 2, the time difference between CLT and the two production lines in scenario 2 is generally lower than that in scenario 1, indicating a better synchronization effect. Without employing the animation simplification method in scenario 1, the time difference even reaches 0.74 s in some cases. The average time difference in scenario 2 has been reduced to 0.05 s by using the animation simplification, which is rarely noticeable by management personnel given that the entire task process lasts approximately 300 s. The simplification of animation resources allows the visualized production line in scenario 2 to retrieve animation resources with less time, resulting in significantly lower time differences in the experimental results compared to scenario 1. The experimental results are presented in this way because scenario 2 has a higher degree of path overlap; the CLT needs to turn around when it needs to travel in different directions on the same route. This result in CLT demonstrates that more time is required to change directions in scenario 2 compared to scenario 1. From this experiment, it is concluded that simplifying animation resources for equipment can enhance the synchronization effect between the two production lines.

Symbol	Description
p_i	The i-th node involved in the production process.
x_{pi}	The x-axis of the i-th node
y_{pi}	The y-axis of the i-th node
t_n	The n-th task.
$ρ_i$	The moving path of the i-th task.
ρ	The hole moving path including {ρ1, ρ2, …, ρn}.
$d(ρ_i)$	The Manhattan distance between the starting and ending nodes of path ρi.
s_{ij}	The length of j-th segment in path ρi.
$ε_i$	The time spent by the equipment in the i-th task phase (phase i).
$ε_i$	The time when the equipment is moving in i.
$μ_i$	The time the equipment is stationary in i.
v_{ij}	The linear velocity at which the device runs on the j-th segment of path ρi.
$θ_{ik}$	The turning angle of the k-th turn in path ρi.
$ω_{ik}$	The angular velocity of the device when turning at the k-th turn of path ρi.The number of line segments in ρi.
a_i	The number of turns in ρi.
b_i	The time when the equipment is moving in i during the previous cycle.
v_{ij}	The time when the equipment is stationary in i during the previous cycle.
$ə_i$	The inherent duration of the animation corresponding to task phase i.
$κ_i$	The speed multiplier of the animation during the playback task phase i.
λi	The start time of phase i written to the database.
$ψ_n$	The end time of phase i written to the database.
$τ_n$	The moment corresponding to the n-th keyframe in the animation.
$α_n$	The moment when the action corresponding to the n-th keyframe is recorded in the database.Way to handle playing animations at the end of the n-th task stage.
$β_n$	The decision result on whether to temporarily accelerate or decelerate the animation when the n-th keyframe arrives.

Table 1. Notations used in this paper.



```
Input: Nodes: $\boldsymbol{p} \leftarrow [p_1, p_2, \ldots, p_n]$, wherein $p_1 \leftarrow (x_{p,1}, y_{p,1})$, $p_2 \leftarrow (x_{p,2}, y_{p,2})$, …, $p_m \leftarrow (x_{p,m}, y_{p,m})$
       Task sequences $\{t_1, t_2, ..., t_n\}$ that specify orders of movements run between nodes
Output: The motion path with the highest degree of overlap $\boldsymbol{\rho}*$, wherein the line segments and
        turning angles that make up each task path: $\{s_{1,1}, \ldots, s_{1,a_1}\}$, $\{\theta_{1,1}, \ldots, \theta_{1,b_1}\}$,
        $\{s_{2,1}, \ldots, s_{2,a_2}\}$, $\{\theta_{2,1}, \ldots, \theta_{2,b_2}\}$, …, $\{s_{n,1}, \ldots, s_{n,a_n}\}$, $\{\theta_{n,1}, \ldots, \theta_{n,b_n}\}$
    The number of line segments and turning angles: $\{a_1, a_2, \ldots, a_n\}$, $\{b_1, b_2, \ldots, b_n\}$
for $i \leftarrow 1$ to $\eta$ do
    randomly generate $\boldsymbol{\rho} \leftarrow \{\boldsymbol{\rho_1}, \boldsymbol{\rho_2}, ..., \boldsymbol{\rho_n}\}$ as in (1)
    for $j \leftarrow 1$ to $n$ do
        divide $\boldsymbol{\rho_j}$ into line segments $\{s_{j,1}, \ldots, s_{j,a_j}\}$ and turning angles $\{\theta_{j,1}, \ldots, \theta_{j,b_j}\}$
        that satisfy (2) and (3)
    end for
    calculate and compare the overlap degrees of the paths, retaining the path
    scheme with the highest overlap degree as (4), (5), (6)
end for
return $\boldsymbol{\rho}*$
```

Algorithm 1. Animation Simplification.

To verify the synchronization effect of CLT between the two production lines at each stage

This section describes the corresponding experiments conducted to verify the synchronization effect of CLT at each stage of the task in the two production lines. To facilitate the timing of each stage of CLT, the task stages are divided, as shown in Fig. 10.

Based on the stages divided in Fig. 10, the operation time of CLT in the simulated production line and the visualized production line was measured separately, and six experiments were conducted, respectively. The same speed has been set for the CLT for the experiments. The differences in settlement time (errors) are shown in Table 3.

	Simulated/Visualized production line time/s						
Sets	1sttime	2ndtime	3rdtime	4thtime	5thtime	6thtime	Average
Set 1.	Sim. Vis. Err.	Sim. Vis. Err.	Sim. Vis. Err.	Sim. Vis. Err.	Sim. Vis. Err.	Sim. Vis. Err.	Sim. Vis. Err.
	297.31 296.62 0.69	296.79 296.43 0.36	297.51 296.97 0.54	296.88 297.53 0.65	297.41 296.98 0.43	297.63 296.89 0.74	297.26 296.90 0.35
Set 2.	300.13 300.02 0.11	299.81 300.11 0.30	300.23 300.04 0.19	300.31 300.13 0.18	299.91 300.19 0.28	300.31 299.91 0.40	300.12 300.07 0.05

Table 2. Measurement of CLT's operating time in the simulated production line and the visualized production line with or without animation simplification. Note that the table has abbreviated"Simulated production line“as”Sim.""Visualized production line“as”Vis.“and”Error“as”Err.".

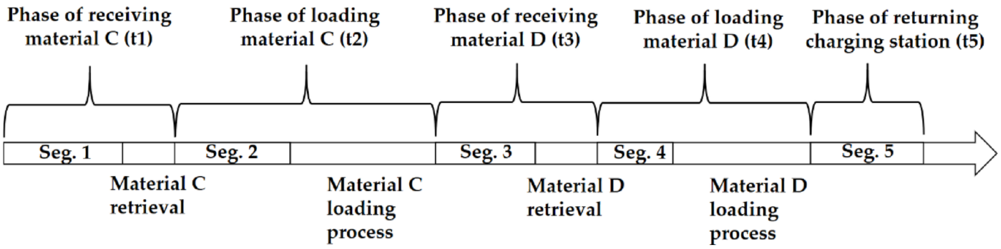

Phase of receiving material C (t1)
Phase of loading material C (t2)
Phase of receiving material D (t3)
Phase of loading material D (t4)
Phase of returning charging station (t5)
Seg. 1
Seg. 2
Seg. 3
Seg. 4
Seg. 5
Material C retrieval
Material C loading process
Material D retrieval
Material D loading process

Fig. 10. The task stages named based on Fig. 8.Note that "Segment" has been abbreviated as "Seg.", and segment 1–5 describes the process of CLT moving on path 1–5.

	Simulated/Visualized production line time/s						
Stage	1 st time	2nd time	3rd time	4 th time	5 th time	6 th time	Average
	Sim. Vis. Err.	Sim. Vis. Err.	Sim. Vis. Err.	Sim. Vis. Err.	Sim. Vis. Err.	Sim. Vis. Err.	Sim. Vis. Err.
Seg.1	30.31 30.59 0.28	30.79 30.81 0.02	30.51 30.59 0.08	30.69 30.78 0.09	30.93 30.84 0.09	30.71 30.52 0.19	30.70 30.69 0.02
Ret. C	16.12 16.02 0.10	16.31 16.19 0.12	16.21 16.08 0.13	16.21 16.07 0.14	16.42 16.31 0.11	16.03 15.94 0.09	16.20 16.10 0.10
Seg.2	26.33 26.31 0.02	26.62 26.51 0.11	26.61 26.63 0.02	26.38 26.58 0.20	26.61 26.42 0.19	26.41 26.52 0.11	26.49 26.50 0.01
Loa. C	71.01 71.11 0.10	71.33 71.31 0.02	71.28 71.49 0.21	71.09 71.11 0.02	71.08 71.12 0.04	71.49 71.61 0.12	71.23 71.29 0.06
Seg.3	26.89 27.02 0.13	27.19 27.12 0.07	27.31 27.09 0.22	27.09 26.92 0.17	27.11 27.23 0.12	26.89 27.11 0.22	27.10 27.08 0.02
Ret. C	6.42 6.19 0.23	6.19 6.18 0.01	6.01 5.82 0.19	5.92 5.73 0.19	5.94 5.92 0.02	6.12 5.89 0.23	6.06 5.96 0.11
Seg.4	23.31 23.21 0.10	23.01 23.22 0.21	23.02 23.04 0.02	23.21 23.08 0.13	23.09 23.03 0.06	23.08 23.20 0.12	23.10 23.13 0.03
Loa. D	59.11 59.18 0.07	59.51 59.41 0.10	59.41 59.58 0.17	59.50 59.41 0.09	59.21 59.29 0.08	59.43 59.52 0.09	59.37 59.40 0.02
Seg.5	40.08 40.32 0.24	40.11 40.09 0.02	39.81 39.79 0.02	40.01 40.09 0.08	40.11 40.22 0.11	39.94 40.03 0.09	40.05 40.09 0.04
Total	299.58 299.95 0.37	301.06 300.84 0.22	300.17 300.11 0.06	300.10 299.77 0.33	300.50 300.38 0.12	300.10 300.34 0.24	300.31 300.23 0.08

Table 3. Measurement of the time difference (error) between the simulated production line and the visualized production line in each stage of CLT's operation. Note that the table has abbreviated "segment" as "seg.", "material C/D retrieval" as "ret. C/D", and "material C/D loading process" as "loa. C/D".

As can be seen from Table 3, in six identical experiments, the time gap between the two production lines for CLT in each stage is generally less than 0.2 s, and the average time difference in the entire process is even lower at 0.08 s.

At the same time, during the testing process, there were no cases where the time difference between the two production lines was extremely large or extremely small. This indicates that through the ASTAK method, not only did the simulated production line and the visualized production line achieve a very high synchronization rate, but their synchronization was also very stable. The data in the table also shows that the time difference between the two production lines for CLT in each stage did not accumulate over time. Attributed to the time alignment method, the animation playback speed of the visualized production line is dynamically adjusted at each stage. This ensures that even if there is a deviation in synchronization between the two production lines in a certain stage, it will not lead to a larger deviation in subsequent stages.

As seen from Table 4, even though the CLT operated at different speeds, the time difference during each stage was still generally below 0.2 s. At the same time, from the time difference for CLT throughout the entire process, it is observed that even when operating at different speeds, the synchronization between the two production lines remains stable. Due to the inclusion of the speeds at various stages as input in the time alignment method, the back-end system is able to consider the changes in the CLT's operational speed when adjusting the animation playback speed. This adjustment approach allows the animation to match the actual operational status of the CLT more closely, thus achieving a better synchronization effect.

Stage	Simulated/Visualized production line time/s						
	1sttime	2ndtime	3rdtime	4thtime	5thtime	6thtime	Average
	Sim. Vis. Err.	Sim. Vis. Err.	Sim. Vis. Err.	Sim. Vis. Err.	Sim. Vis. Err.	Sim. Vis. Err.	Sim. Vis. Err.
Seg.1	30.81 30.78 0.03	26.71 26.53 0.18	23.67 23.48 0.19	21.02 20.91 0.11	18.52 18.32 0.20	16.03 15.85 0.18	22.79 22.65 0.14
Ret. C	16.33 16.19 0.14	16.21 16.13 0.08	16.03 16.04 0.01	16.33 16.22 0.11	16.01 16.03 0.02	16.22 16.23 0.01	16.17 16.14 0.03
Seg.2	26.62 26.69 0.07	23.33 23.48 0.15	20.78 20.89 0.11	18.33 18.51 0.18	15.82 15.85 0.03	13.43 13.32 0.11	19.73 19.79 0.06
Loa. C	71.29 71.32 0.03	70.89 71.02 0.13	71.52 71.61 0.09	71.22 71.44 0.22	70.73 70.95 0.22	71.18 71.21 0.03	71.14 71.26 0.11
Seg.3	27.23 27.31 0.08	23.48 23.59 0.11	20.92 20.93 0.01	18.42 18.51 0.09	15.96 16.04 0.08	13.51 13.52 0.01	19.93 19.98 0.05
Ret. C	6.22 6.08 0.14	6.08 5.99 0.09	6.11 5.93 0.18	6.33 6.15 0.18	5.82 5.65 0.17	6.23 6.04 0.19	6.11 5.97 0.14
Seg.4	23.03 23.19 0.16	20.62 20.73 0.11	18.10 18.02 0.08	15.72 15.55 0.17	13.33 13.14 0.19	10.92 10.93 0.01	16.98 16.93 0.05
Loa. D	59.51 59.41 0.10	59.21 59.11 0.10	59.33 59.4 0.08	59.32 59.23 0.09	58.93 59.15 0.22	59.04 59.18 0.14	59.21 59.25 0.04
Seg.5	40.19 40.13 0.06	34.83 34.92 0.09	30.32 30.39 0.07	26.91 27.03 0.12	23.54 23.55 0.01	20.12 20.28 0.16	29.31 29.38 0.08
Total	301.23 301.10 0.13	281.36 281.50 0.14	266.78 266.70 0.08	253.60 253.55 0.05	238.66 238.68 0.02	226.68 226.56 0.12	261.36 261.35 0.02

Table 4. Measurement of the time difference (error) between the simulated production line and the visualized production line in each stage of CLT's operation when its running speed varies.

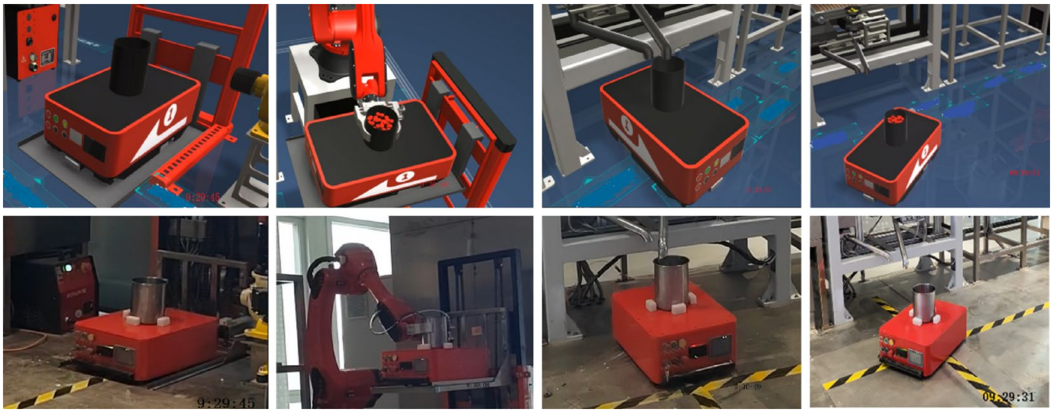


Fig. 11. CLT action in the visualized production line and the simulated production line.

CLT maintains consistent instantaneous actions at all stages

Through the keyframe synchronization method introduced in the Keyframe Synchronization section, instantaneous action synchronization of CLT between the simulated production line and the visualized production line during the production process has been achieved. Examples of instantaneous action synchronization in CLT are provided in Fig. 11. As shown in Fig. 11, at the same system time, the actions of CLT in the visualized production line and the simulated production line are completely consistent, which is reflected not only in its consistent position on the production line but also in its consistent posture.

Note that the images of CLT in the simulated production line and the visualized production line were captured from different camera positions. The image of CLT in the simulated production line was captured by a handheld camera with a relatively low height, while the image of CLT in the visualized production line was captured by a camera built into the system with a relatively high height. Therefore, there is a certain difference in the observational turning angle of CLT in the images. However, in actual operation, CLT has achieved the best instantaneous synchronization effect between the two production lines.

The experiments conducted on the operation of CLT in the simulated production line and the visualized production line have demonstrated that the ASTAK method enhances the visual experience of the equipment running on the visualized production line. It aids managers on actual industrial production lines in better understanding the operational status of the equipment.

Discussion

This article introduced the ASTAK method and conducted extensive experiments to verify its effectiveness in achieving synchronization between equipment in simulated (or actual) production lines and visualized production lines. There are several points remaining for future investigation regarding the synchronization of equipment between actual production lines and visualized production lines:

- Providing animation resources for equipment malfunctions that enable managers to interpret the production line's status. This animation resource will address the difficulty of understanding the current situation during equipment malfunctions.

- Improving the communication efficiency between PLC, database, and visualized production line back-end systems further reduces synchronization errors. The existence of a database increases the burden of communication between production lines, while improving communication efficiency will reduce this impact.

Conclusion

To improve the synchronization effect between the simulated (or actual) production line and the visualized production lines in industrial production, this article proposes an ASTAK method. The ASTAK method integrates three main innovations, including animation simplification, time alignment, and key frame synchronization, to address the synchronization challenges between simulated and visualized production lines. Through extensive experiments, the ASTAK method has been verified to reduce the time difference (error) between the simulated production line and the visualized production line to 0.08 s. To the best of our knowledge, the state-of-the-art methods only achieve a synchronization rate of 94.75%. However, our method achieves a synchronization rate of 99.97%, indicating a significant increase of 5.22%. The ASTAK method not only improves the synchronization rate of equipment between simulated and visualized production lines, but also reduces the impact of deviations on synchronization. Additionally, this ASTAK method enables managers to better understand the operating status of equipment in the production line, facilitating the improvement of management efficiency for staff in unmanned factories.

Data availability

Data is provided within the manuscript or supplementary information files.

Received: 13 May 2024; Accepted: 21 April 2025
Published online: 25 April 2025

References

1. Xu, Q., Wu, F., Xiang, R., Zhou, J. & Yang, X. *Autonomous steel casting recognition and positioning on the unmanned automatic production line based on binocular vision*. Vol. 11197 111970USPIE, (2019).
2. Kreuzer, T., Papapetrou, P. & Zdravkovic, J. Artificial intelligence in digital twins—A systematic literature review. *Data Knowl. Eng.* **151**, 102304 (2024).
3. Mohsen, S., Behrooz, A. & Roza, D. Digital twin for smart manufacturing, A review. *Sustainable Manuf. Service Econ.* **2**, 100017 (2023).
4. Turgay, S., Akar, N. J. D. M. & Management, P. Digital twin modeling and simulation of computer aided design and manufacturing structure: case study. *Digit. Manuf. Process. Manage.* **3**, 1–10. https://doi.org/10.23977/dmpm.2023.030101 (2023).
5. Židek, K., Piteľ, J., Adámek, M., Lazorík, P. & Hošovský, A. Digital twin of experimental smart manufacturing assembly system for industry 4.0 concept. *Sustainability* **12**, 102304. https://doi.org/10.3390/su12093658 (2020).
6. Zongyan, W. in *Industry 4.0* (eds Bányai Tamás & Felice Antonella Petrilloand Fabio De) Ch. 7IntechOpen, (2020).
7. Botín-Sanabria, D. M. et al. Digital twin technology challenges and applications: A comprehensive review. *Remote Sens.* **14**, 1335. https://doi.org/10.3390/rs14061335 (2022).
8. Wu, W. J. & Ye, G. J. Optimization of elevator production line in its flexible improvement by the value stream mapping tool. *Appl. Mech. Mater.* **442**, 233–237. https://doi.org/10.4028/www.scientific.net/AMM.442.233 (2014).
9. Wu, P. et al. in *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC).* 875–879.
10. Jiang, X. et al. Design of simulation verification platform for ship sub assembly digital production line control system. *J. Phys: Conf. Ser.* **1650**, 032167 (2020).
11. Qiu, H., Zhang, H., Lei, K., Zhang, H. & Hu, X. Forest digital twin: A new tool for forest management practices based on Spatio-Temporal data, 3D simulation engine, and intelligent interactive environment. *Comput. Electron. Agric.* **215**, 108416. https://doi.org/10.1016/j.compag.2023.108416 (2023).
12. Kim, H. W., Cho, M. & Lee, M. C. Image processing techniques for improving quality of 3D profile in digital holographic microscopy using deep learning algorithm. *Sensors* **24**, 1950 (2024).
13. Bret, J., F, K. D. & Lift-Off Using reference imagery and freehand sketching to create 3D models in VR. *IEEE Trans. Vis. Comput. Graph.* **22**, 1442–1451 (2016).
14. Yin, J. & Xiao, D. in *The 2011 International Conference on Advanced Materials and Information Technology Processing(AMITP 2011).* 6.
15. Liang, P. & Xin-han, H. in *The 2011 International Conference on Advanced Materials and Information Technology Processing(AMITP 2011).* 7.
16. Xing, Y. et al. Hybrid synchronisation method based on inverse generalised and inverse projected high dimensional discrete chaotic systems. *Phys. Scr.* **99** (2024).
17. Xinyi, L. et al. in *6th International Workshop on Advanced Algorithms and Control Engineering (IWAACE 2022).*
18. Yiping, K. et al. Neurosurgeon:Collaborative intelligence between the cloud and mobile edge. *ACM SIGOPS Operating Syst. Rev.* **51**, 615–629 (2017).
19. Holger, Z. Real-Time-Capable synchronization of digital twins. *IFAC PapersOnLine.* **54**, 147–152 (2021).
20. Sangmi, P., Changhee, H., Inkyu, H. & Jaewook, L. Comparison of Single-Camera-Based depth Estimation technology for digital twin model synchronization of underground utility tunnels. *Appl. Sci.* **13**, 2106–2106 (2023).
21. Kampa, A. Modeling and simulation of a digital twin of a production system for industry 4.0 with Work-in-Process synchronization. *Appl. Sci.* **13**, 12261 (2023).
22. Lee, T. Y., Lin, C. H., Wang, Y. S. & Chen, T. G. Animation Key-Frame extraction and simplification using deformation analysis. *IEEE Trans. Circuits Syst. Video Technol.* **18**, 478–486 (2008).
23. Shixue, Z., Jinyu, Z. & Bin, W. in *2nd International Conference on Computer Engineering and Technology.* V1-681-V681-685. (2010).
24. Huang, B. S., Shen, D. F., Lin, G. S. & Chai, S. K. D. in *2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS).* 136–139.
25. Krejsa, J. & Liarokapis, F. in *IEEE Conference on Games (CoG).* 1–9. (2021).
26. Guo, G., Wang, Y., Zhang, H. F. & Li, S. in *International Conference on Industrial Informatics, Machinery and Materials(IIMM 2015).* 51–54. (2015).
27. Pechter, W. H. Synchronizing keyframe facial animation to multiple Text-to-Speec. *Dartm. Coll. Undergrad. Theses*, **38** (2004).
28. Freire, I. et al. Testbed evaluation of distributed radio timing alignment over ethernet fronthaul networks. *IEEE Access.* **8**, 87960–87977. https://doi.org/10.1109/ACCESS.2020.2993204 (2020).

29. Jung, C. & Noh, T. Study on Three-Dimensional Curved-Surface machining using industrial articulated robot. *Eng. - Trans. Korean Soc. Mech.* **35**, 1071–1076 (2011).
30. Zhuge, M., Li, X. & Liang, C. Research on Vehicle Temperature Regulation System Based on Air Convection Principle *IOP Conference Series: Materials Science and Engineering* 322, 062022 (2018).
31. Wang, H., Yang, Z., Zhang, Q., Sun, Q. & Lim, E. A digital twin platform integrating process parameter simulation solution for intelligent manufacturing. *Electronics* **13**, 802 (2024).
32. Valencia-Palomo, G. & Rossiter, J. A. Efficient suboptimal parametric solutions to predictive control for PLC applications. *Control Eng. Pract.* **19**, 732–743 (2011).
33. Wall, D. Yaesu FT3DR Dual-Band analog and digital handheld transceiver. *QST* **104**, 38–42 (2020).
34. Gao, P., Wen, J. & Hu, Y. in *2014 IEEE 7th Joint International Information Technology and Artificial Intelligence Conference.* 550–554.
35. Salahuddin, S., Widdha, M. & Azman, A. Perancangan robot Laba-Laba Pendeteksi api berbasis mikrokontroller ATMEGA32. *Elkawnie* **4**, 79–88 (2018).

Acknowledgements

This research achievement has received support and cooperation from the Yantai Huangbohai New Area Administration Committee and Baidu.com Technology Co.

Author contributions

Professor Yue Dong conceived the experiments, Zixiao Wang and professor Shengguo Wang conducted the experiments, Dr. Xinxiang Zhang and Zixiao Wang analyzed the results. Dr. Xinxiang Zhang organized the modification of the report. All authors reviewed the manuscript.

Funding

This study received no external funding.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-025-99483-x.

Correspondence and requests for materials should be addressed to Z.W.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.