Construction and Simulation of a Virtual Judging Process Model for the Fosbury Flop Based on Priority-Constrained TCPN

Pengfei Wen[1], Dongbin Lai[1], Yuan Zheng[2], Zhi Li[1], Jinsong Wu[1†]

[1]School of Physical Education, Guangzhou Sport University, Guangzhou 510500, Guangdong, China
[2]School of Physical Education and Sports Science, South China Normal University, Guangzhou 510006, Guangdong, China

Keywords: Fosbury flop; Virtual judging; Timed colored petri Net; Priority constraint matrix; Simulation

Citation: Wen P.F., Lai D.B., Zheng Y., et al.: Construction and Simulation of a Virtual Judging Process Model for the Fosbury Flop Based on Priority-Constrained TCPN. Data Intelligence, Vol. 7, Art. No.: 2025r45, pp. 1147–1168, 2025. DOI: https://doi.org/10.3724/2096-7004.di.2025.0060

ABSTRACT

This study proposes a virtual judging system for the Fosbury Flop based on the Timed Colored Petri Net (TCPN) and a Priority Constraint Matrix (PCM), aiming to enhance the accuracy and real-time capabilities of event judging. First, an in-depth analysis of the judging requirements for the Fosbury Flop was conducted, defining key judgment nodes such as takeoff, bar clearance, and landing. A Petri net model was used to model the timing and state of each node. The system employs a PCM to control the execution order of judgment nodes, ensuring logical consistency and automation in the judging process. In simulation testing, the system's judgment results in various scenarios—such as normal jumps, time violations, bar contact infractions, and unstable landings—were consistent with manual judgments, demonstrating high adaptability, stability, accuracy, and reliability. The results indicate that the virtual judging system can automatically recognize athlete actions and make judgments, effectively reducing human errors and providing innovative technical support for referee training and event management. Future research could incorporate more auxiliary resources to further optimize the system's response speed, as well as integrate machine learning algorithms to enhance the intelligence and applicability of the judging system.

† Corresponding author: Jinsong Wu (E-mail: 11047@gzsport.edu.cn).

© 2025 Chinese Academy of Sciences. Published under a Creative Commons Attribution 4.0 International (CC BY 4.0) license.

1. INTRODUCTION

The Fosbury Flop, a groundbreaking high jump technique, has been the mainstream technique for high jump events since the 1968 Mexico City Olympics. This technique, invented by Dick Fosbury, allows athletes to jump backward over the bar, keeping their center of gravity below the bar, thereby increasing their clearance height [1]. Although the technique provides athletes with a significant advantage in jumping height, the complexity of the movements makes judging more challenging, requiring referees to accurately assess the athletes' dynamic performance of the takeoff, bar clearance, and landing phases [2]. At present, most of the judging systems of high jump events still mainly rely on artificial vision and limited video assistance, so it is difficult for the referees to make accurate judgment of the movement details in the process of rapidly changing movement, so there is a risk of misjudgment [3].

With advancements in athletic event technology, virtual judging systems have gradually been adopted across various sports to reduce the misjudgment caused by human factors. For instance, World Athletics has introduced technological video replay equipment in recent years for sprints and relay events, enhancing the accuracy and fairness of judging through multi-angle video monitoring. Nevertheless, the application of traditional systems in high jump events remains limited, relying primarily on whether athletes touch the bar as the main criterion, making it difficult to handle the fast dynamic actions involved in the Fosbury Flop [3].

The Petri net, a classical modeling tool for concurrent systems, has been widely used for process control and dynamic behavior analysis of complex systems. Its graphical structure allows for an intuitive representation of system states and their changes, making it particularly suitable for controlling complex processes involving concurrency and synchronization. In the field of sports officiating, the Petri net can be used to simulate and analyze the dynamic changes in multi-step judgment processes, ensuring precise control of each judgment node through their clear state and event descriptions [4]. Timed Colored Petri Net (TCPN) further expands its functionality, realizing the distinction between event time series and resources by incorporating time and color marking, then the TCPN becomes an ideal tool for simulating complex athletic systems [4]. Studies have shown that the TCPN can effectively capture important temporal information and conditions during athletic events, providing detailed judging criteria for referees and reducing errors caused by referees' subjective judgments [5]. In the Fosbury Flop judging process, there is a strong coupling between the temporal constraints of motion phases and penalty priorities. While TCPN can characterize event time attributes, it struggles to directly represent motion phase priority logic. The Priority Constraint Matrix (PCM) defines mandatory state transition sequences, ensuring penalty decisions align with priority logic. This enables the Fosbury Flop virtual referee system to maintain rational penalty decisions even under complex motion overlaps.

Developing a virtual judging system for the Fosbury Flop has significant value. First, such a system can achieve real-time monitoring and automatic judgment of athlete movements through high-precision video analysis and modeling algorithms, to reduce human misjudgment and omissions, and ensure fairness and consistency in judging. Second, the automation features of a virtual judging system can enhance event

operational efficiency and reduce labor demands, especially in large-scale competitions where it can significantly alleviate the workload and management complexity for referees. Additionally, this system can be used for referee training, serving as a support tool to help referees enhance their judging skills in a simulated environment to provide long-term technical support for event management.

The purpose of this study is to construct a virtual judging process model for the Fosbury Flop using the TCPN and the PCM. By modeling and simulating judgment nodes of referees, this model allows the system to monitor and provide automatic judgments in real-time across different movement phases (e.g., takeoff, bar clearance, landing). Through the research and development of this model, the virtual judging system can not only offer accurate judgment support in actual competitions but also be applied to referee training, helping referees become familiar with the judging standards and procedures of the Fosbury Flop. This system provides technical support and theoretical basis for future event management and referee training.

2. CONSTRUCTION OF THE VIRTUAL JUDGING PROCESS MODEL

To achieve precise judgments for each movement node in the Fosbury Flop judging process and to ensure sequential and consistent operations, this study employs a modeling approach based on the priority-constrained TCPN. The Petri net is a mathematical representation of discrete parallel systems and is suitable for describing relationships such as sequence, concurrency, coexistence, and competition [6]. The following are the specific steps and definitions used in constructing the model.

Definition 1: Timed Colored Petri Net [7]

Timed Colored Petri Net is an extended Petri network model that introduces time and color attributes to describe the discrete parallel events and their temporal relationships in a system. A TCPN can be represented as $TCPN = (P, T, F, C, K, W, M_0, R)$, where the components are defined as follows:

(1) P is a limited set of places, which represent the various judging states in the system. In the context of the Fosbury Flop virtual judging system, the place P represents the different states in the judging process, such as "Valid Takeoff", "Successful Bar Clearance", "No Fault in Landing", and so on.

(2) T is a limited set of transitions, representing the changes between judging states. For example, a transition T might represent the state transition from "Bar Clearance Judgment" to "Landing Judgment".

(3) F is a set of directed arcs, which defines the relationships between places and transitions. $F \subseteq (P \times T) \cup (T \times P)$, where places and transitions connected by directed arcs represent the dependencies between judging nodes.

(4) C is a color set, which assigns colors to each place and transition to differentiate between various judging resources or conditions (e.g., video replays, timers, etc.). The colors help distinguish judging nodes that require different resources during the judging process.

(5) K is a capacity function for places, defined as $K:P\rightarrow\mathbb{N}$, which defines the maximum number of tokens that each place can hold. In the judging model, capacity represents resource limitations for a particular judging state, such as the number of times a video replay resource can be used.

(6) W is a weight function on arcs, defined as $W:F\rightarrow\mathbb{N}$, which defines the consumption of resources along the arcs. The weight function represents the consumption of resources by different judging actions. For example, some transitions might consume one video replay resource.

(7) M_0 is an initial marking vector, which represents the marking state of each judging node of the system in the initial state. $M_0:P\rightarrow\mathbb{N}$, specifies the initial number of marking for each place $p \in P$.

(8) R is a time set, defined as $R \subseteq \mathbb{R}_{\geq 0}$, which defines the time constraints for each transition. For example, the trigger time of transition T might be defined as the time limit between the takeoff and bar clearance in the Fosbury Flop.

Definition 2: Priority-Constraint Timed Colored Petri Net [8]

The Priority-Constraint Timed Colored Petri Net (PTCPN) is a model that adds priority constraints to the TCPN, suitable for modeling of judging discrete events in scenarios with sequential dependencies. PTCPN can be represented as $PTCPN = (P, T, F, C, K, W, M_0, R, Pd, S, S_0)$, where:

(1) P, T, F, C, K, W, M_0, R: Retains the same definition as the TCPN, representing the set of places, the set of transitions, the set of directed arcs, the color set, the place capacity function, the weight function, the initial marking vector, and the time set.

(2) Pd is the decision place set $Pd \subset P$, used to determine whether the triggering conditions are met at each judging node. Decision places are special places used to assess the availability or state of current judging resources. For example, if the takeoff judging node in the Fosbury Flop judging system is determined to be "valid", it will trigger the decision place for the next judging node of bar clearance.

(3) S is the PCM $S = (s_{ij})_{n \times n}$, used to represent the priority relationships between judging nodes. Each element s_{ij} of matrix S indicates the priority constraint between node i and node j, defined as follows:

① $s_{ij} = 0$ indicates that transitions t_i and t_j are not adjacent, meaning no direct priority relationship exists.

② $s_{ij} = 1$ indicates that transition t_i occurs before t_j.

③ $s_{ij} = -1$ indicates that transition t_i occurs after t_j.

(4) S_0 is the initial PCM S_0, used to represent the priority relationships of the system at the beginning. The initial definition of the priority relationships ensures the logical sequence of the judging process. For example, the initial priority constraint relationships of the takeoff judging node is typically set before the bar clearance judging node. The time constraints of the Petri net can be expressed as:

$$\text{Triggering conditions of } t_i = \begin{cases} M(p) \geqslant W(p,\ t_i), & \forall p \in P \\ \text{Time constraint satisfaction meets the conditions}, & r_t - r_{t_i 0} \geqslant r_{t_i} \end{cases}$$

Definition 3: Reverse Operation

If the transition t_i in the judging process represents a state change of a certain judging node (e.g., from "valid takeoff" judgment to "bar clearance" judgment), the reverse operation t_i^{-1} will be defined as the opposite direction transition of the state of the node. For example, if the bar clearance is judged as successful, but after replaying the takeoff, it is judged as a foul, the reverse operation t_i^{-1} needs to be triggered, which is referred to as $t_i \cap t_i^{-1} = \varnothing$.

Definition 4: Priority Constraints of Forward and Reverse Operations

Assume that a complete set of transitions in the referees' judging process is denoted as $R = \{t_1, t_2, \ldots, t_n\}$, and the PCM is $S_{RA} = (s_{ij})_{n \times n}$. The set of transitions of the corresponding reverse judging process is denoted as $I = \left\{t_1^{-1}, t_2^{-1}, \ldots, t_n^{-1}\right\}$, and the PCM of reverse operations is $S_{IA} = (s_{ij}^{-1})_{n \times n}$, which represents the reverse priority relationships between transitions t_i and t_j. In the context of the Fosbury Flop judging model, if there is a need to backtrack a particular action (such as re-judging the takeoff), the model can gradually backtrack the states of the related transitions according to the reverse PCM, ensuring the accuracy of the judgment result.

Definition 5: Trigger Conditions of a Transition

The conditions for the triggering of transition t_i under marking M are as follows:

(1) $\forall p_1 \in {}^*t_i \Rightarrow M(p_1) \geqslant W(p_1, t_i) \wedge \forall p_2 \in t_i{}^* \Rightarrow M(p_2) + W(t_i, p_2) \leqslant K(p_2)$.

(2) $r_t - r_{t_i 0} \geqslant r_{t_i}$.

In other words, when the condition of a judging node is satisfied (e.g., the athlete successfully takes off and does not cross the line), the transition is triggered, and the judging node progresses to the next state.

Definition 6: State Update After Transition Trigger

For $\forall p \in P$, if transition t has the firing right under marking M, after it is triggered:

(1) Update the marking M':

$$M'(p) = \begin{cases} M(p) - W(p,\ t), & p \in {}^*t - t^* \\ M(p) + W(t,\ p), & p \in t^* - {}^*t \\ M(p) - W(p,\ t) + W(t,\ p), & p \in t^* \cap {}^*t \\ M(p), & p \notin t^* \cup {}^*t \end{cases}$$

(2) Delete the rows and columns corresponding to the transition t_i in the PCM S, indicating that the transition has been completed, and the process moves to the next judging node.

3. OPERATIONAL META-MODEL OF THE FOSBURY FLOP VIRTUAL JUDGING

In the virtual judging system, to accurately judge the various movement states of athletes and precisely model the process of the judging actions, it is necessary to modularize the integration of judging actions, auxiliary resources, and the final model. Specific symbols and diagrams are defined to model the interaction processes between various judging actions and auxiliary resources within the virtual judging system, and the meta-model is used to achieve process integration. The following provides a detailed explanation and modeling of each component.

3.1 Modeling of Judging Behavior

Judging behavior modeling involves defining various judging states (e.g., valid attempt, invalid attempt, and foul) and the conditions for state transitions, forming an operational meta-model to ensure the scientific accuracy and consistency of the judging process.

Valid Attempt: Denoted as A_{valid} or O, this state represents that the athlete completes the takeoff, bar clearance, and landing within the stipulated time, without any violations such as bar contact or falling. When the valid conditions are met in the judging process, the system transitions to this state.

Invalid Attempt: Denoted as $A_{invalid}$ or X, this state represents that the athlete fails to successfully complete the action or commits errors such as knocking down the bar during the bar clearance or landing. In such cases, the system transitions to the invalid state.

Foul: Denoted as A_{foul} or P, this state represents that the athlete commits a foul, such as exceeding the time limit or taking off with both feet. When the foul conditions are met, the system transitions to this state.

The state transitions of judging behavior can be represented using the matrix M_A, which defines the transition relationships among different judging states:

$$M_A = \begin{bmatrix} A_{start} & \rightarrow & A_{valid} & O \\ A_{start} & \rightarrow & A_{invalid} & X \\ A_{start} & \rightarrow & A_{foul} & P \end{bmatrix}$$

Explanation: When the jump initiation node is judged to be valid, the process proceeds to the "bar clearance" judging node. If the judgment result is invalid or foul, the process terminates immediately, and the result is recorded.

3.2 Modeling of Judging Auxiliary Resources

To ensure the continuity and accuracy of the judging process, auxiliary resources such as timers (T), video replay systems (V), and triggers (S) are introduced into the model.

3.2.1 Timers (T)

Timers are used to monitor the time limit for each takeoff. If the athlete exceeds the time limit without completing the takeoff, the system automatically judges it as a foul P.

T_{start}: Timer start, indicating the athlete is called to take off.

T_{end}: Timer end, triggering a foul judgment if the time limit is exceeded.

The timer state transition matrix is defined as:

$$M_T = \left[T_{start} \rightarrow T_{end} \quad \text{Trigger } P \text{ on timeout}\right]$$

3.2.2 Video Replay System (V)

The video replay system is used to record jump details and support secondary judgments. At critical judging nodes, video resources can be utilized to ensure judgment accuracy. The video resource only intercepts the key frames of the action (e.g., the crossbar area at the moment of crossing the pole), and blocks the background and irrelevant people through image cropping technology to avoid capturing the auditorium or other sensitive information.

V_{replay}: When judging bar clearance, video replay resources are called to ensure accurate evaluation.

The video replay invocation matrix is defined as:

$$M_V = \left[P_{clearance} \rightarrow V_{replay} \rightarrow \text{Judge as } O / X\right]$$

3.2.3 Triggers (S)

Triggers are used to detect the athlete's takeoff position, bar clearance height, and landing position, providing real-time support for judgment. The data collected by the trigger retains only the movement parameters (e.g., touch signal strength, jump time) and does not record the athlete's physical characteristics (e.g., body type, skin color, etc.).

$S_{takeoff}$: Detects takeoff action, determining whether the takeoff is single-footed and within the designated area.

$S_{clearance}$: Detects bar clearance conditions, determining whether the bar is touched.

$S_{landing}$: Detects landing actions, ensuring the athlete lands safely.

The trigger state matrix is defined as:

$$M_S = \begin{bmatrix} S_{takeoff} & \rightarrow & P_{start} \\ S_{clearance} & \rightarrow & P_{clearance} \\ S_{landing} & \rightarrow & P_{end} \end{bmatrix}$$

The combined auxiliary resource symbol matrix is:

$$M_R = \begin{bmatrix} T_{start} & T_{end} & V_{replay} & S_{takeoff} & S_{clearance} & S_{landing} \end{bmatrix}$$

3.3 Fusion of the Meta-model

After constructing the judging behavior model and auxiliary resource model, these modules are integrated into a unified judging process through meta-model fusion. The meta-model fusion primarily includes three components: judging nodes, auxiliary resources and state recording.

3.3.1 Synchronized Triggering of Judging Nodes

In the meta-model fusion, the execution order and time window of each penalized node is controlled by the timing constraints of TCPN and the prioritization logic of PCM. Each node's triggering is influenced by the combined action of timers, video replay systems and triggers, with all interactions recorded in the state matrix, the details are as follows:

(1) Timing-driven: the time set R of TCPN defines the maximum allowed time for each node. For example, the trigger of the starting node P_{start} needs to satisfy $T_{start} \leqslant T_{max,\,start}$, otherwise it triggers the reverse variation t_1^{-1}, which is directly judged as a timeout foul P.

(2) Resource dependency: auxiliary resources (timer T, video replay V, trigger S) are assigned to the corresponding libraries by color marking $C(p)$. For example, the pole-crossing node $P_{clearance}$ needs to call the video replay resource $V_{\text{replay}} \in C(P_{\text{clearance}})$ with the capacity constraint $K(P_{\text{clearance}}) = 2$.

(3) State transitions: the trigger order of the penalized nodes is defined by the matrix M_P:

$$M_P = \begin{bmatrix} P_{start} \rightarrow T_{start} \rightarrow S_{takeoff} \rightarrow A_{valid} \,/\, A_{invalid} \,/\, A_{foul} \\ P_{clearance} \rightarrow V_{replay} \rightarrow S_{clearance} \rightarrow A_{valid} \,/\, A_{invalid} \\ P_{end} \rightarrow S_{landing} \rightarrow A_{valid} \,/\, A_{invalid} \end{bmatrix}$$

The PTCPN simulation diagram illustrates the complete process, including judging nodes, resource invocations, and state transitions, and the token flow path is consistent with the manual referee logic, which is as shown in Figure 1.

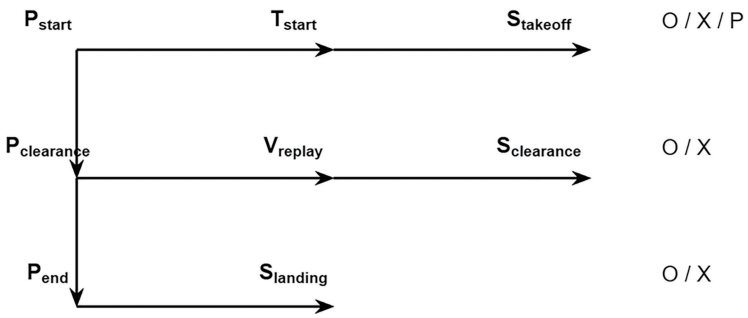


Figure 1. Flow Diagram of PTCPN Simulation in the Fosbury Flop Judging Process.

3.3.2 Data Recording and Updates

During the judging process, the system records the state of each action and the auxiliary resources invocations in real-time. This data is saved in the judging data matrix D. The data matrix recorded by the system contains only action times, penalty results and number of resource calls, and does not involve athlete identification information (e.g., name, facial features, biometric data).

$$D = \{D_{time},\ D_{status},\ D_{resource}\}$$

D_{time}: Time record, indicating the execution time of each node.

D_{status}: Judgment results (e.g., *O, X, P*).

$D_{resource}$: Usage of auxiliary resources (e.g., *T, V, S*).

4. IMPLEMENTATION PROCESS OF REFEREES' JUDGING BASED ON PTCPN

To standardize the referees' judging process, a complete judgment workflow was constructed using the TCPN and the PCM. The PCM controls the triggering sequence of judgment nodes, while the TCPN provides time control and resource management. The following describes the specific implementation process.

4.1 Jump Initiation Node Judging Process

The jump initiation node is the starting point of the Fosbury Flop process, responsible for detecting whether the athlete completes the takeoff within the stipulated time. This node has the highest priority, ensuring that the judging process begins with the takeoff phase.

Judging Conditions and Steps:

Timer Activation: When the athlete enters the takeoff position, the system activates the timer T, monitoring the takeoff time window.

Time Constraint Judgment: If the takeoff time is within the stipulated limits, it is judged as compliant and proceeds to the next judging node; if the athlete's takeoff time T_{start} exceeds the maximum allowable time $T_{max,\ start}$, the system automatically judges it as a "timeout foul". The judgment formulas are expressed as follows:

If $T_{start} \leqslant T_{max,\ start}$, then P_{start} = Compliant (O)

If $T_{start} > T_{max,\ start}$, then P_{start} = Timeout Foul (P)

Execution of the Judging Process: If the result is compliant, the process moves to the next node; if the result is a timeout foul, the process terminates, and the jump is recorded as invalid.

4.2 Bar Clearance Node Judging Process

The bar clearance node follows the jump initiation node. At this node, the system detects whether the athlete successfully clears the bar without touching it, ensuring compliance with the judging rules.

Judging Conditions and Steps:

Sensor and Video Replay Activation: When the athlete enters the bar clearance phase, the sensor S and video replay system V are activated to monitor the athlete's bar clearance actions in real time.

Bar Clearance Judging Conditions: If the athlete clears the bar without touching it, the system marks it as "Compliant (O)"; if the athlete touches the bar during clearance, the system marks it as "Foul (X)". The judgment formulas are expressed as follows:

If $S_{clearance} = 0$, then $P_{clearance}$ = Compliant (O)

If $S_{clearance} = 1$, then $P_{clearance}$ = Foul (X)

Where $S_{clearance} = 0$ indicates no bar contact, $S_{clearance} = 1$ indicates bar contact.

Execution of the Judging Process: If the bar clearance judgment is compliant, the process moves to the landing node; If it is a foul, the process terminates, and the jump is recorded as invalid.

4.3 Landing Node Judging Process

The landing node is the final judging node, responsible for detecting the athlete's actions and positions during landing, ensuring compliance with the event's landing standards.

Judging Conditions and Steps:

Boundary Detection and Sensor Activation: When the athlete completes the bar clearance and enters the landing phase, the boundary detection system B and sensor S are activated to monitor the athlete's landing stability and location.

Landing Judging Conditions: If the athlete lands within the designated area and the landing action is stable, the system marks it as "Valid (O)"; If the athlete lands outside the designated area or performs an unstable landing, the system marks it as "Invalid (X)". The judgment formulas are expressed as follows:

If $B_{landing} = 0$, then P_{end} = Valid (O)

If $B_{landing} = 1$, then P_{end} = Invalid (X)

Where $B_{landing} = 0$ indicates compliance with the designated area, and $B_{landing} = 1$ indicates a violation outside the area.

Execution of the Judging Process: If the landing judgment is valid, the jump is recorded as a valid jump (O); if it is invalid, the process terminates, and the jump is recorded as invalid (X).

Each place and transition correspond to the judging nodes and the control conditions, forming the process flow, which is as shown in Figure 2.

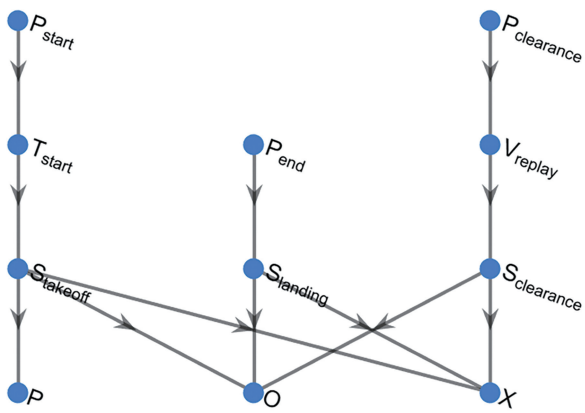


Figure 2. Flow Diagram of the Priority-Constrained Referees' Judging Process.

5. EXAMPLE SIMULATION OF JUDGING THE FOSBURY FLOP

The example simulation section primarily demonstrates the judgment process of the virtual judging system across different scenarios. By constructing a PCM and TCPN model, the system's judgment process is simulated step-by-step. The following illustrates two main scenarios based on the characteristics of the Fosbury Flop: a successful jump and a foul jump.

5.1 Key Transitions and Their Meanings in the Judging Process

In the Fosbury Flop judging process, different Transitions represent various judging actions. Based on the key elements of the high jump process, the transitions are defined, which is as shown in Table 1.

Table 1. Forward Transition Symbols and Judging Actions.

Transition Symbols	Judging Actions
t_1	Activate the timer and start the countdown for takeoff.
t_2	Judge whether the athlete completes the takeoff within the stipulated time.
t_3	Check whether the athlete performs a single-foot takeoff.
t_4	Judge whether the bar clearance is successful.
t_5	Judge whether the athlete lands safely.
t_6	Record the jump result (successful or failed).

In reverse transitions, we define actions that may require re-judgment. For example, if the initial judgment marks the athlete as having touched the bar, but a review reveals no contact, a reverse transition can be used to correct the judgment, which is as shown in Table 2.

Table 2. Reverse Transition Symbols and Judging Actions.

Transition Symbols	Judging Actions
t_1^{-1}	Stop the timer and restart the countdown for takeoff.
t_2^{-1}	Rejudge whether the takeoff was completed within the allowed time.
t_3^{-1}	Reconfirm whether the takeoff action complies with the rules.
t_4^{-1}	Recheck the bar clearance result.
t_5^{-1}	Rejudge the validity of the landing action.

5.2 Key Places and Their Meanings in the Judging Process

Places are used to represent the athlete's state at each phase of the judging process. The places for the high jump judging process are defined, which is as shown in Table 3.

Table 3. Place Symbols and Their Meanings.

Place Symbol	Meaning
p_{10}	The athlete has received the takeoff notification.
p_{20}	The athlete is in position to takeoff preparation.
p_{30}	The athlete has completed the takeoff.
p_{40}	The athlete has cleared the bar.
p_{50}	The athlete has completed the landing action.
p_{60}	The jump result has been recorded.

These places indicate different states in the judging process. The system monitors the status of each place and performs state transitions according to the Transitions defined in Table 2, updating the judging process in real time.

5.3 Simulation Scenarios for Fosbury Flop Judging

Scenario 1: Successful Jump

In this scenario, the athlete completes the takeoff, bar clearance, and landing actions as required, and the judging system records the jump as "valid". The process is as follows:

(1) Takeoff Phase (Transitions t_1, t_2 and t_3):

The timer T starts. The athlete receives the take-off notification. The place status is "Received Takeoff Notification" p_{10}. The athlete reaches the take-off preparation position. The place status transitions from "Received Takeoff Notification" p_{10} to "Ready to Takeoff" p_{20}. The athlete completes the takeoff within the stipulated time. The place status transitions from "Ready to Takeoff" p_{20} to "Takeoff Completed" p_{30}, and the system records the judgment as valid (O).

(2) Bar Clearance Phase (Transition t_4):

The system detects that the athlete successfully clears the bar without touching it. The place status transitions from "Takeoff Completed" p_{30} to "Bar Clearance Completed" p_{40}, and the system records the judgment as valid (O).

(3) Landing Phase (Transition t_5 and t_6):

The athlete lands safely. The place status transitions from "Bar Clearance Completed" p_{40} to "Landing Completed" p_{50}. The jump result has been recorded. The place status transitions from "Landing Completed" p_{50} to "Recording Completed" p_{60}, and the system records the jump result as successful (O).

Flow Diagram:

$$p_{10} \xrightarrow{t_1} p_{20} \xrightarrow{t_2} p_{30} \xrightarrow{t_3} p_{40} \xrightarrow{t_4} p_{50} \xrightarrow{t_5} p_{60} \xrightarrow{t_6} \text{Judgment Result: O}$$

Scenario 2: Foul Jump

In this scenario, the athlete fails to complete the takeoff within the stipulated time, thus the system judges it as a timeout foul. The process is as follows:

(1) Takeoff Phase (Transition t_1 and t_1^{-1}):

The timer T starts. The athlete receives the take-off notification. The place status is "Received Takeoff Notification" p_{10}. The athlete reaches the take-off preparation position. The place status transitions from "Received Takeoff Notification" p_{10} to "Ready to Takeoff" p_{20}. But the athlete fails to complete the takeoff

within the allowed time. This triggers the reverse transition t_1^{-1}, and the place status remains at "Ready to Takeoff" p_{10}, and failed to be converted to "Takeoff Completed" p_{30}. The system directly judges it as a foul (P).

(2) Process Termination:

Since the takeoff phase is judged as a timeout foul, the system skips the bar clearance and landing nodes, and directly records the jump result as invalid.

Flow Diagram:

$$p_{10} \xrightarrow{t_1} p_{20} \xrightarrow{t_1^{-1}} \text{Judgment Result: P}$$

5.4 Judging Matrix and Simulation Diagram

To further illustrate the example processes, the following judging matrix and Petri net simulation diagram are provided:

(1) Judging State Matrix ($M_{Penalty}$):

The judging matrix is used to record the athlete's status and judgment results at each stage.

$$M_{Penalty} = \begin{bmatrix} p_{10} & t_1 & p_{20} & t_2 & p_{30} & t_3 & p_{40} & t_4 & p_{50} & t_5 & p_{60} \\ O & X & P & & & & & & & & \end{bmatrix}$$

Among it, O represents a valid jump. X represents an invalid jump. P represents a foul.

(2) Petri Net Simulation Diagram:

The state transitions and resource invocations in the process are illustrated by the Petri net simulation diagram, which is as shown in Figure 3.

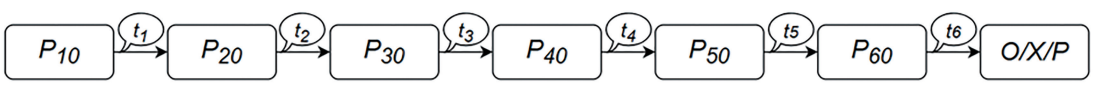


Figure 3. Petri Net Simulation Diagram.

This Petri net model demonstrates the system's dynamic judging process under different conditions, helping to understand the execution mechanism of the virtual judging system in the Fosbury Flop.

5.5 Virtual Judging System Simulation Tests

This section verifies the reliability and accuracy of the Fosbury Flop virtual judging system through multi-dimensional data. For this purpose, the study designs various simulation test scenarios, including normal jumps, boundary conditions, and abnormal conditions. By incorporating diverse data, such as

different athletes, environmental conditions, and judging errors, the simulation tests are made more comprehensive and rigorous.

5.5.1 Simulation Test Design

The simulation test scenarios for evaluating the system's performance under different conditions are defined, which is as shown in Table 4.

Table 4. Test Scenarios and Judgments.

Test Scenario	Description	Expected Judgment Result
Normal Jump Test	The athlete completes the jump as required, with the takeoff, bar clearance, and landing actions all meeting the standards.	Valid Jump (O)
Timeout Jump Test	The athlete fails to complete the takeoff within the allowed time.	Timeout Foul (P)
Bar Contact Foul Test	The athlete touches the bar during clearance, causing the bar to fall.	Foul Jump (X)
Unstable Landing Test	The athlete lands unstably or out of bounds after clearing the bar.	Invalid Jump (X)
Continuous Jump Test	Multiple athletes perform consecutive jumps, requiring the system to handle sequential judgments.	Individual Results

5.5.2 Multi-dimensional Data Recording and Analysis

(1) Penalty data for different scenario tests

In the jump tests for different athletes, the system recorded the response times and judgment results of each athlete to ensure applicability under different scenarios, which is as shown in Table 5.

Table 5. Athlete Test Results and Judgments.

Athlete ID	Test Scenario	Takeoff Time (s)	Bar Clearance Time (s)	Landing Time (s)	Judgment Result
A001	Normal Jump	50	1.3	1.8	O
A002	Timeout Jump	67	-	-	P
A003	Bar Contact Foul	43	1.4	-	X
A004	Unstable Landing	48	1.1	2.6	X

(2) Penalty data for different jump heights

Jumps at different heights were tested to assess the system's adaptability in penalizing difficult maneuvers, which is as shown in Table 6.

Table 6. Penalties for different height tests.

Athlete ID	Test Scenario	Jump height (m)	Judgment Result
A001	Normal Jump	1.8	O
A002	Timeout Jump	1.9	P
A003	Bar Contact Foul	2.0	X
A004	Unstable Landing	1.7	X
A005	Continuous Jump (1st Jump)	1.8	O

(3) Penalty data for repeated testing of the same test scenario

To verify the consistency of the system's penalty results, several tests were conducted under the same conditions, which is as shown in Table 7.

Table 7. Multiple tests and penalties.

Test Scenario	Number of tests	Response time (s)	Judgment Result
Normal Jump	1	7.2	O
	2	7.1	O
	3	7.3	O
Timeout Jump	1	5.5	P
	2	5.6	P
	3	5.4	P

(4) Penalty Error and Accuracy Data

To quantify the system's judging accuracy, the system's judgment results were compared with manual judgments, and the errors and accuracy for each test were calculated, which is as shown in Table 8.

Table 8. Judgment Errors and Accuracy Across Test Scenarios.

Test Scenario	System Judgment Result	Manual Judgment Result	Judgment Error	Accuracy (%)
Normal Jump	O	O	0	100
Timeout Jump	P	P	0	100
Bar Contact Foul	X	X	0	100
Unstable Landing	X	X	0	100
Continuous Jump (1st)	O	O	0	100

(5) Referee's subjective rating data for the system

Professional referees evaluated the accuracy and timeliness of the system's judgments for each scenario to obtain subjective assessment data, which is as shown in Table 9.

Table 9. Expert Evaluation Results.

Test Scenario	Accuracy Score (1-5)	Timeliness Score (1-5)	Overall Evaluation
Normal Jump	5	5	Excellent Performance
Timeout Jump	5	4	Good Performance
Bar Contact Foul	4	4	Correct Judgment
Unstable Landing	5	5	Accurate Judgment
Continuous Jump (1st)	5	4	Stable and Reliable

5.5.3 Summary of Simulation Test Results

Multi-dimensional data demonstrate that the system performs stably and accurately under various test scenarios, environmental conditions, and individual differences. This is manifested in the following areas:

(1) Adaptive: The system can complete the judgment within the specified time under different test scenarios, and different jumping difficulties, with timely response and good adaptability.

(2) Stability: In multiple tests under identical conditions, the system produced consistent judgment results, verifying its stability.

(3) Accuracy: The system's penalization accuracy in all test scenarios is 100%, and is completely consistent with the results of manual penalization.

(4) Reliability: Professional referees gave high ratings to the system's accuracy and timeliness, further supporting its reliability.

6. CONCLUSIONS AND PROSPECTS

6.1 Research Conclusions

This study constructed a virtual judging system for the Fosbury Flop based on the PTCPN. Through the modeling and simulation of the judging process, an automated and standardized judging procedure was achieved. The main conclusions of this study are as follows:

6.1.1 Implementation of Efficient Judging Process

By using the PCM and Petri net modeling, the judging system is capable of dynamically controlling the execution order of various judging nodes, thus effectively ensuring the logicality and consistency of the Fosbury Flop judging process.

6.1.2 Implementation of Automated Judging Process

By adopting the TCPN model, the judging conditions and time constraints were integrated into the judging process, achieving the automated judging of key actions such as takeoff, bar clearance, and landing. This reduces the need for human intervention and enhances the efficiency of the judging.

6.1.3 Data Recording and Traceability of Judging

The system records the judgment results of each jump in real-time. These data not only provide the basis for the referee's decisions but also serve as valuable references for future training and event analysis, offering good traceability and transparency.

6.2 Research Prospects

Although this study has realized the preliminary modeling and simulation of the virtual judging system for the Fosbury Flop, there are still areas for improvement and future research directions:

6.2.1 Applicability and Generalizability of the Model

In the future, this model can be applied to the judging systems of other athletics events to further verify its applicability across various sports scenarios. By adjusting the parameters of the Petri net and the judging conditions, the model could be made more adaptable to a wider range of events.

6.2.2 Introduction of Intelligent Algorithms

Building on the current model, machine learning and deep learning algorithms can be incorporated to automatically recognize and analyze athletes' movements and postures, improving the accuracy of judging and reducing the rate of misjudgments.

6.2.3 Multi-resource Fusion Optimization

In the future, more auxiliary resources such as high-precision sensors and real-time image processing systems could be integrated with existing video replay and timer resources. This would provide more comprehensive support for the judging process, further enhancing the reliability and accuracy of the system.

6.2.4 Real-time Performance and Response Speed Optimization of the System

Given the real-time requirements in practical applications, future research could focus on optimizing the computational efficiency of the Petri net model to improve the system's response speed, ensuring it can meet the real-time judging demands of large-scale events.

In summary, this study provides preliminary theoretical and technical support for the automation and intelligence of the Fosbury Flop virtual judging system. Future research and improvements will contribute to the generalization of the system in practical applications and promote the modernization and intelligent development of sports judging technologies.

AUTHOR CONTRIBUTIONS

Pengfei Wen (11419@gzsport.edu.cn) and Jinsong Wu (11047@gzsport.edu.cn) were responsible for the design of the framework and the analysis of the results. Dongbin Lai (1244881469@qq.com) and Yuan Zheng (zy702900402@163.com) was responsible for the data and model calculations of the article, and Zhi Li (11752118@qq.com) optimized the language and format of the manuscript.

ACKNOWLEDGEMENTS

This study is supported by the Educational Science Planning of Guangdong Province project of China (Grant Number: 2023GXJK354), and the Higher Education Teaching Research and Reform of Guangdong Province project of China in 2024.

REFERENCES

[1] E. S. Hintz. Accessed: Nov. 2024. *The Fosbury Flop—A Game-Changing Technique FosburyFlop.* [Online]. Available: https://invention.si.edu/invention-stories/fosbury-flop-game-changing-technique.

[2] IAAFNEWS. Accessed: Nov. 2024. *New way to judge jumps take-off fouls among recent competition rule changes introduced for 2020 and 2021.* [Online]. Available: https://worldathletics.org/news/news/competition-rules-changes.

[3] J. L. Peterson, "Petri Nets Theory and the Modelling of Systems," *Prentice Hall,* 1983.

[4] R. Wolfgang, "Understanding petri nets-modeling techniques," *Analysis Methods, Case Studies. Springer,* 2013.

[5] J. W. Herrmann and E. Lin, "Petri Nets: Tutorial and Applications," in *The 32th Annual Symposium of the Washington Operations Research-Management Science Council, Washington, DC*, 1997.

[6] Y. Huang, X. Hu, G. Lv, and R. Yang, "A simulation modeling method based on Petri net," in *2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing*, 2015, pp. 283–288: IEEE.

[7] D. Lefebvre and E. Aguayo-Lara, "Initial study for observers application to fault detection and isolation with continuous timed petri nets," *IFAC-PapersOnLine,* vol. 48, no. 7, pp. 97–103, 2015.

[8] Y. Wu, et al. "Solving resource-constrained multiple project scheduling problem using timed colored Petri nets," *Journal of Shanghai Jiaotong University (Science),* vol. 14, pp. 713–719, 2009.

AUTHOR BIOGRAPHY



Pengfei Wen was born in 1987, Hebei Province, P.R. China. He received the Doctor's degree from Hebei Normal University, P.R. China. Now working in School of Physical Education of Guangzhou Sport University, Guangzhou,China, Dedicated to research on Exercise and Health Promotion. Teaching and training of track and field.
E-mail: 11419@gzsport.edu.cn.



Dongbin Lai was born in 2000, Jiangxi Province, P.R. China. He studied at the School of Physical Education of Guangzhou Sport University, P.R. China, Dedicated to research on Exercise and Health Promotion, Sport and Artificial Intelligence.
E-mail: 1244881469@qq.com.



Yuan Zheng was born in 2001, Hunan Province, P.R. China. She studied at the School of Physical Education and Sports Science of South China Normal University, P.R. China, Dedicated to research on Exercise and Health Promotion.
E-mail: zy702900402@163.com.



Zhi Li was born in 1982, Hubei Province, P.R. China. He received the Master degree from the Wuhan Sports University, P.R. China. Now working in School of Physical Education of Guangzhou Sport University, Guangzhou, China, Dedicated to research on Physical Education.
E-mail: 11752118@qq.com.



Jinsong Wu was born in 1968, Hubei Province, PR. China. He received the Master degree from Guangzhou Sport University, P.R. China. Now working in School of Physical Education of Guangzhou Sport University, Guangzhou, China, Dedicated to research on Teaching and training of track and field.
E-mail: 11047@gzsport.edu.cn.