Development of a stress-free algorithm for controlling active running platforms

Artem Obukhov[1*], *Sergey* Karpushkin[2], *Aleksandr* Siukhin[2], *Kirill* Patutin[1], and *Yaroslav* Averin[1]

[1]Department of Automated Systems of Decision-Making Support, Tambov State Technical University, Tambov, 392000, Russian Federation
[2]Department of Computer Integrated Systems in Mechanical Engineering, Tambov State Technical University, Tambov, 392000, Russian Federation

Abstract. The problem of increasing the accuracy of predicting human actions is an urgent task for various human-machine systems. The study examines the solution to the problem of predicting human speed using neural network algorithms, computer vision technologies, and machine learning. The formalization and software implementation of a neural network speed prediction algorithm are presented. To solve the problems of determining the current speed and predicting the upcoming positions of the human body depending on the dynamics of its movement, a comparison of various machine learning models was carried out. The RandomForestRegressor algorithm showed the best position prediction accuracy. The best determination of the current speed was demonstrated by dense multilayer neural networks. The experiment revealed that when predicting a person's position at an interval of 0.6 seconds, his speed is determined with an accuracy of more than 90%. The results obtained can be used to implement neural network algorithms for controlling human-machine systems.

1 Introduction

Active running platforms, which are commonly known as controlled treadmills, are frequently used devices for simulating physical activity. They are successfully used for rehabilitation in medicine, professional training of personnel in the mining industry. Such software and hardware platforms make it possible to simulate the terrain and control the position of the user's body in real time, as well as test his physical condition.

A distinctive feature of active-running platforms is a closed cycle of control reactions. This cycle is presented as follows: the user makes an external influence on the platform, on the basis of which the control system (CS) generates a response, which also affects the user. Thus, the control process is the adaptation of the running platform's response to the external environment (user actions).

The process of generating the response is based on obtaining and analyzing data about the user's movement, which is carried out in real time. The main problem is the lag, since some time passes from the moment the user's current position begins to change until the

* Corresponding author: obuhov.art@gmail.com

corresponding response of the running platform begins to be implemented. The reason for the lag is the sequential execution of several functions by the CS: recording the position and speed of the user's movement; calculating the inertia of running platform components; and determining limitations based on safety requirements and user conditions. The duration of performing these functions creates a response lag of the treadmill, which prevents smooth real-time control.

Lag in the CS of active running platforms is the main source of negative sensations that cause stress. Thus, the user is distracted from the process of interacting with virtual reality, which negatively affects the performance of the tasks assigned to him. To eliminate this problem, it is necessary to develop new, stress-free algorithms for controlling the active running platforms, which is the aim of this study.

An analysis of research on this topic has shown that currently many running platform control algorithms demonstrate a significant level of lag. The summary analysis is presented in Table 1.

Table 1. Comparison of machine learning algorithms using the MAE metric.

Study	Lag of the running platform, s	User speed before stopping, m/s	User displacement after stopping, m
Hedjazi N. [1]	0.43-0.57	1.2	0.52-0.64
Lee H. [2]	0.85	3.0	2.50
Lichtenstein L. [3]	1.48	1.3	1.92
DeLuca A. [4]	2.00	0.5	1.00
Pyo S. [5]	2.00	2.5	5.00

Thus, there is the lag of at least 0.43 seconds, which is not enough for natural and stress-free movement while the speed of movement changes. With a total length of the treadmill of 1–1.5 meters, this can lead to a fall. Constant changes in the speed of the running platform negatively affect the vestibular apparatus of the user, which forces the rehabilitation or training process to stop. The negative effect is significantly enhanced if the user trains in a virtual reality helmet, i.e., does not visually control his real position on the simulator.

Collecting data about user actions for decision making may require the use of additional equipment that limits freedom of movement. Therefore, an urgent task is the development of algorithms that make it possible to predict the user actions and provide compensation for the lagged response of the running platform, which is understood in this paper as the stress-free algorithm. By analyzing and processing incoming information for the purpose of comparison with existing movement patterns, it is possible to predict and respond to user actions in time.

To collect information about gait, various devices called observers (mechanical and ultrasonic sensors, three-dimensional positioning devices, and video cameras) are used. As a result of the analysis of their features [6–8], the following requirements are formulated: the absence of attachments on the user that restrict his freedom of movement; observer insensitivity to the background electromagnetic radiation of the treadmill; observer coverage ≥ 1.5 m (typical length of running platform). An analysis of existing motion tracking systems [9] showed that only video cameras meet these requirements. However, their use requires compliance with a condition: the user's body must be clearly different from the background. This problem can be solved using modern machine learning models, for example, MediaPipe, MoveNet, PoseNet, and more advanced algorithms [10]. However, it must be taken into account that the use of these models leads to additional computational costs and, therefore, may increase the lag of the CS.

2 Materials and methods

Based on the analysis, it is concluded that the development of stress-free algorithms for controlling the active running platforms using computer vision technologies and machine learning models is necessary to minimize the lag of the feedback response.

The concept of the proposed stress-free algorithm is that it is necessary to generate and analyze arrays of user movements (obtained after processing by machine learning models) and then select the appropriate action from the library of control commands that is most suitable within the current state of the person. To eliminate the lag, analysis and decision-making is carried out not only based on a set of previous frames but also taking into account the forecast of further human actions by classifying and predicting his movements.

Next, the basic designations of objects obtained as a result of a subject area system analysis of the control process using a system of cameras and computer vision are introduced:

- D is a set of elements in the database of recognized human movements;
- D_i is a sequence of user movements from the database;
- Θ is the CS lag time that needs to be minimized;
- $Data$ is a set of data from cameras;
- $data_{ij}$ is a frame from sequence D_i, containing an array of recognized points of the human body;
- $data_t$ is a current frame from camera.

The control process of running platforms with observers is based on transmitting current $data_t$ frames to the CS directly and changing the speed of the platform according to the user's displacement in each frame, as was done in a previous study [11]. In the implementation of the stress-free algorithm, the predicted values of the user's body points are transferred to the CS, which allows partially or fully compensating for the lagged response of the running platform while the user is moving.

To select a specific array of repeated user movements and the corresponding frames from cameras, it is necessary to classify and predict the further trajectory of movement and compare them with the database. If a comparable trajectory is not found in the database, then control is carried out according to current data in standard mode.

Thus, the following is a statement of the research problem. Based on the results of the analysis of the current frame $data_t$ of the user's position, select from database D a sequence of repeated movements $D_i \in D$. When selecting a specific movement $d_{ij} \in D_i$ from a given sequence and transferring it to the CS instead of the current movement from frame $data_t$, the minimum response lag of the running platform is:

$$\Theta(D_i) \rightarrow min, \tag{1}$$

$$d_{ij} = \left\{\left\langle data_{ij}, \left[min\left|data_{ij}\right|, max\left|data_{ij}\right|\right]\right\rangle\right\}, \tag{2}$$

where $D_i = \left\{d_{ij}\right\}$, $j = 1,...,J_i$;

$data_{ij}$ is an array of points in the human body corresponding to d_{ij};

$\left|data_{ij}\right|$ is a characteristic of the frame $data_{ij}$. The characteristic of the frame can be understood as the position of the key points of a person in it (for example, the averaging or central point), which determines his movement or position;

$\left[min\left|data_{ij}\right|, max\left|data_{ij}\right|\right]$ is the range of permissible values of the characteristics of frame $data_{ij}$ of element d_{ij} of array D_i. If the characteristic $\left|data_t\right|$ of the current frame $data_t$ satisfies the region $\left[min\left|data_{ij}\right|, max\left|data_{ij}\right|\right]$, then $data_t$ can be replaced by $data_{ij}$:

$$\left|data_t\right| \in \left[min\left|data_{ij}\right|, max\left|data_{ij}\right|\right]. \tag{3}$$

Using frame characteristics makes it possible to minimize the influence of arbitrary user movements; perfect repetition of movements is not required, and minor deviations are acceptable (for example, up to 5%).

The algorithm for solving problems (1)–(2) includes the following operations: processing input frames; forming a motion database; and selecting array D_i for transmission to the CS.

Processing frames from observers includes removing defects, determining the characteristics of frames, and forming an array of pairs, including recorded arrays of data on the position of the user's body key points and the corresponding values of numerical characteristics during T frames.

$$Data = \left\{\left\langle data_t, \left| data_t \right| \right\rangle\right\}, t = 1,...,T. \tag{4}$$

The formation of arrays of repeated user movements begins with identifying patterns in the sequences of elements of array $Data$. Next, the number J_m of frames in each of them is determined and the fulfillment of the conditions is checked:

$$J_m \geq \Theta_0 \cdot v, m \in [1,...M], \tag{5}$$

$$M > 1, \tag{6}$$

where Θ_0 is the initial response lag of the active running platform. At the moment of receiving frame $data_t$ the platform reacts to frame $data_{t-\Theta_0 \cdot v}$, received Θ_0 seconds earlier. The CS generates the response to the received frame $data_t$ only after Θ_0 seconds;

v is the rate of transmission of user frames by cameras to the CS, fps;

$\Theta_0 \cdot v$ is the amount of lag, frames;

M is the number of identified patterns (if a single pattern corresponds to any value J_m, then it is excluded from consideration since one frame is not enough to identify a correspondence between the current sequence of frames and the recorded frames in the database).

Patterns corresponding to conditions (5) and (6) are grouped based on the number of frames. Thus, the following n elements of a set of J_n elements are formed:

$$P_n = \left\{\left\langle data_{nj}, \left| data_{nj} \right| \right\rangle\right\}; n = 1,...,N_n; j = 1,...,J_n; N_n > 1. \tag{7}$$

Afterwards, a group of patterns is selected, including the maximum number of frames from the array $Data$, which corresponds to the maximum of criterion Q:

$$Q = N_n \cdot J_n / L, \tag{8}$$

where N_n is the number of patterns of user movement with the same number J_n, L is the number of frames in array $data_{nj}$.

From the values of the frame characteristics of the selected group of patterns P_n the areas of their permissible values are formed:

$$\left[\min_{n=1,...N_n}\left|data_{nj}\right|, \max_{n=1,...N_n}\left|data_{nj}\right|\right]; j = 1,...,J_n. \tag{9}$$

Then a single pattern is identified based on the criterion of the maximum average value of the characteristics of their frames:

$$\sigma_n = \frac{1}{J_n}\sum_{j=1}^{J_n}\left|data_{nj}\right|; n = 1,...,N_n. \tag{10}$$

From the selected pattern, an array $D_i \in D$ is formed and placed in the database. All frames with user body positions included in this array are excluded from the array $Data$ and a new iteration is carried out to identify patterns. The iterations are repeated until it is possible to form patterns that satisfy formulas (5) and (6).

The selection of the array $D_i \in D$ from the database, a certain frame of which is then transferred to the CS instead of the real frame from observers $data_t$, is carried out according to (3). If this condition is met for the initial frames of several arrays D_i, the fulfillment of condition (3) is checked for subsequent frames until a single array is determined.

Thus, it is necessary to form a database and select from it an array D_i of repeated user movements, transferring a certain frame $data_{ij}$ of element d_{ij} to the CS instead of a real frame $data_t$ to compensate for the lag of user actions.

The selection of a specific element of the array D_i, the frame $data_{ij}$ of which is then transmitted to the CS instead of the real frame $data_t$ from observers, is carried out when executing (4) based on the result of determining the maximum possible compensation time for the treadmill's response lag:

$$\Theta_i = \Theta_0 - (J_i - t_i) / v, \tag{11}$$

where t_i is the number of array elements D_i used in the process of its selection.

If $\Theta_i > 0$, then transferring a frame of any element of the selected array to the CS can only partially compensate for the lagged response of the running platform. A similar situation arises if the data transfer rate from observers is insufficient and the hardware and software are not optimized. In this situation, the last frame of the D_i array is transmitted to the simulator CS:

$$\Theta_i > 0 \Rightarrow data_{t,J_i}. \tag{12}$$

If $\Theta_i \leq 0$, then transferring to the CS a frame of one of the selected array elements D_i can completely compensate for the response lag to user actions:

$$\Theta_i \leq 0 \Rightarrow data_t \rightarrow data_{i,c}. \tag{13}$$

where $c = \Theta_0 \cdot v + t_i$ is the number of the element of array D_i. As a result, when transferring a frame from array D_i to the CS, the lag is completely compensated. Full compensation provides an instant response to the user's movements; partial compensation does not provide an instant response but increases the safety of the process of using the active running platform and reduces stress levels.

3 Results and dscussion

The following describes the procedure for applying the stress-free algorithm. Figure 1 shows the stages of processing data from a video sequence: recognizing the human body in the frame; comparing the characteristics of the current frame with the database; finding the closest dataset in terms of characteristics; and sending the predicted movement to the CS. The original frame is processed by the MoveNet model to extract human body key points, which are then matched to a database. To test the software modules that implement the above-described algorithm, video streams of user training is used, filmed at a camera frame rate of $v = 30$ fps.

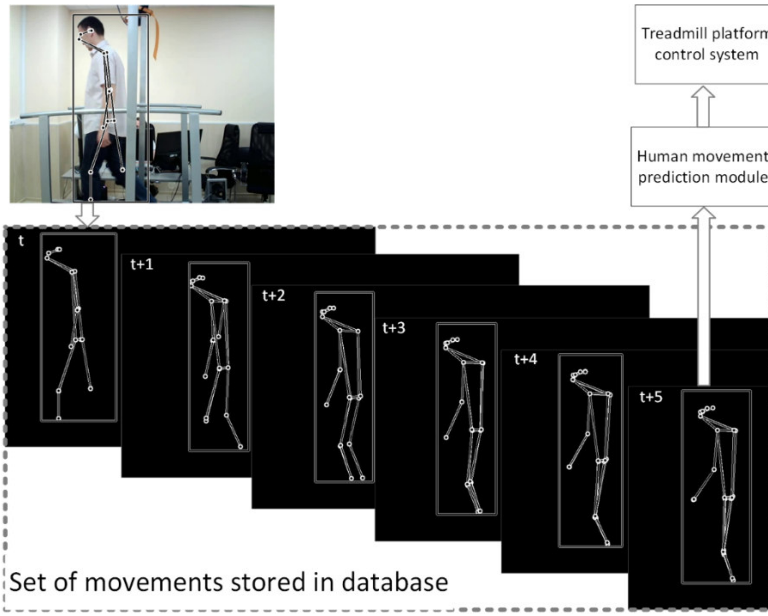

Fig. 1. Execution of the stress-free algorithm, including the stages of processing the initial frame and selecting an array of movements from the database.

In the current study, to unambiguously select an array of repeated user movements, it is necessary to check the fulfillment of condition (4) for $t = 5$ consecutive user positions on frames (0.167 s). The maximum lag compensation time, according to (12), is:

$$\Theta = \Theta_0 - (29-5)/30 = \Theta_0 - 0.8. \tag{14}$$

Therefore, if Θ_0 corresponds to the minimum value from table 1 (0.43 seconds), the proposed method for predicting user movements can fully compensate for the lag. Transmitting frame $data_{i,c}$ where $c = 0.43 \cdot 30 + 5 = 13$ instead of the current frame $data_t$ from observers compensates for the response lag by 100%. If Θ_0 corresponds to the maximum value from table 1 (2 seconds), then it can be reduced to $\Theta = 1.2$ seconds: transferring frame $data_{i,29}$ to the CS instead of $data_t$ compensates for the response lag by 40%.

4 Conclusion

The results of this study are listed below. The task of controlling a running platform was posed, the solution of which allowed for compensation for the lag. The stress-free platform control algorithm was developed based on the recognition and systematization of user movements and the comparison of the current frame with a database of typical movements. As a result of the algorithm's operation, an array of user movements was selected from the database, and then a frame of a certain element was transferred to the CS, which made it possible to partially or completely compensate for the response lag of the running platform.

Testing of software modules that implement algorithms using a video stream of a specific user's training showed that in 0.167 s (at a frame transmission rate of observation cameras of 30 fps), it is possible to predict his movements for a period of up to 0.8 s. As a result, full compensation is provided for the minimum lag of the CS $\Theta_0 = 0.43$ s and partial compensation for the maximum lag $\Theta_0 = 2$ s.

The research was carried out at the expense of the grant of the Russian Science Foundation No. 22-71-10057, https://rscf.ru/en/project/22-71-10057/.

References

1. N. Hedjazi, A. Benali, M. Bouzit, Z. Dibi, *An omnidirectional platform design: application to posture analysis,* in XIV Mediterranean Conf. on Medical and Biological Engineering and Computing, MEDICON 2016, 31st March - 2nd April 2016, Paphos, Cyprus (2016)
2. H. Lee, S. Pyo, S. Park, J. Yoon, *Design of the omnidirectional treadmill based on an omni-pulley mechanism*, in 13th Int. Conf. on Ubiquitous Robots and Ambient Intelligence, URAI, 19-22 August 2016, Xian, China (2016)
3. L. Lichtenstein, J. Barabas, R.L. Woods, E. Peli, ACM T. Appl. Percept. **4**, 1-23 (2007)
4. A. De Luca, R. Mattone, P.R. Giordano, H. Ulbrich, M. Schwaiger, M. Van den Bergh, L. Van Gool, IEEE T. Contr. Syst. T. **21**, 410-427 (2012)
5. S. Pyo, H. Lee, J. Yoon, Appl. Sci. **11**, 4223 (2021)
6. J.L. Souman, P.R. Giordano, I. Frissen, A.D. Luca, M.O. Ernst, ACM T Appl. Percept. **7**, 1-14 (2010)
7. T. Kikuchi, K. Sakai, K. Ishiya, J. Robot. Mechatron. **27**, 528-534 (2015)
8. M. Auralius, J.W. Yoon, J. Inst. Control Robot. **17**, 505-511 (2011)
9. A. Obukhov, D. Dedov, A. Volkov, D. Teselkin, Comput. **11**, 85 (2023)
10. J.L. Chung, L.Y. Ong, M.C. Leow, Future Internet **14**, 380 (2022)
11. A.D. Obukhov, M.N. Krasnyanskiy, D.L. Dedov, V.V. Vostrikova, D.V. Teselkin, E.O. Surkova, Neural. Comput. Appl. **34**, 12919-12946 (2022)