PID-Driven Autonomous Motorcycle Jump with rider effect

Wenjia Li

[#]*Integrative Engineering, Lafayette College*

730 High St, Easton, PA 18042

[1]liwe@lafayette.edu.edu

***Abstract*—Motorcycle and bicycle jumps are increasingly recognized as a high-risk feature in sports, especially as the popularity of motorcycle and bicycle racing has surged in recent years. The dynamics of powered two-wheeler (PTW) jumps share similarities with those of ski jumps. While there is extensive research on ski jumps, studies focused on PTW jumps remain scarce. Additionally, due to the inherent risks of conducting experiments with human participants, existing research often lacks robust simulation data. This project aims to answer the following question: Can we design and simulate how the rider's movement affects the bike's behavior during a jump in simulation?**

***Keywords*— Motorcycle, bicycle, PTWs, ski, jumps, PID, rider.**

I. Introduction

Motorcycle and bicycle jumps are essential features in sports such as supercross, motocross, BMX, and mountain biking, and have become increasingly popular among athletes and spectators alike. However, these activities carry significant risks. In 2022, 98 motorcycle riders were killed during off-road riding [1], and motorcycle jumps are among the most injury-prone activities in off-road sports. Attempting unfamiliar or complex jumps for the first time can be especially dangerous, as various critical factors—such as takeoff speed, landing angle, takeoff trajectory, and rider technique—greatly influence both safety and performance.

While extensive research has been conducted on ski jumps [2], one of the key metrics identified is the Equivalent Fall Height (EFH). EFH is related to the impact kinetic energy generated by the component of landing velocity perpendicular to the landing surface[2]. In ski jump design, it is crucial to consider a limited, constant EFH when determining the takeoff angle and shaping the landing surface, as this helps to achieve a safer jump design [3], [4], [5], [6], [7].

Despite extensive research on ski jumps, there is a significant gap when it comes to studying motorcycle and bicycle jumps, with only one paper found specifically about bicycle jumps[8]. During the previous work, a PTW jump was designed in simulation, however, it doesn't include the rider's movement influence. So for this project, I will aim to answer the question: whether it is possible to study the rider's movement influence during the jump in simulation?

The structure of this paper is as follows: Section II provides a review of relevant studies on ski jumps and powered two-wheeler (PTW) jumps. Section III outlines the methodology employed to achieve the project's objectives. Section IV presents the simulation results, while Section V offers a discussion and Section VI concludes the paper and highlights potential future work. Finally, Sections VII include the references.

II. Literature Review

Recognizing the similarities in dynamics between motorcycle and ski jumps, such as the speed, trajectory, and impact forces—are shared across both sports, this literature review will incorporate relevant studies on ski and PTW jumps.

This literature review focuses on sports jumps specifically using search strings to find papers on jumps for motorcycles,bicycles,scooter, or ski. From a total of 50 articles found in IEEE Xplore, including 44 conference papers, 4 magazines and 2 journal papers. After the title screening, there are 28

papers left and after full paper screening 15 relevant papers were selected for further analysis. Also an extended search from google scholar where 5 papers are selected. So they're totally 20 papers involved for further analysis.
Notably, there is only one paper specifically addressing PTW jumps [8], while the remaining studies focus on ski jumps. In the following sections, I will provide a more detailed review of the relevant literature.

A. ski jumps

In [9], [10], [11], [12], and [13], the authors use terrestrial magnetism and acceleration sensors, combining these with camera video to measure the flying distance. This approach allows for a comprehensive analysis of jump dynamics, as the sensors provide quantitative data on acceleration and movement, while the camera video also offers insights into the landing shock. However, despite the effectiveness of the camera systems and sensors in capturing jumps, this methodology necessitates human involvement in the experimental process. Athletes must perform the jumps, which inherently carries risks and dangers.

Since the takeoff phase in ski jumping is crucial for achieving optimal performance, the authors in [14] utilized a synchronized camera system to capture this critical moment. This technology aids in training new jumpers by providing visual feedback on their technique during takeoff, allowing them to make necessary adjustments for improved performance. Paper [15] introduces a smart, compact, and energy-efficient wireless sensor system designed for real-time performance analysis and biofeedback in ski jumping. The system aims to enhance the training process by providing athletes and coaches with immediate data on performance metrics, enabling them to make informed adjustments to technique and strategy. In [16] and [17], a camera system is used to detect the ski jumper’s pose. While these methods are convenient for guiding novice jumpers, they also pose inherent risks. Allowing inexperienced athletes to attempt ski jumps can be dangerous due to the high speeds and technical precision required.

In [18] and [19], the authors designed wearable inertial-magnetic sensors to measure the ski jump landing momentum. In [20], an image processing-based method was introduced to analyze the height and width of an athlete's jump, differing from existing sensor-based approaches. Calculations for vertical height and horizontal distance are based on recorded video from a fixed distance, using the Euclidean distance formula. In [21], fuzzy logic is used in sports jump detection to filter out noise and obtain more accurate accelerometer and GPS data. In [22], inertial sensor devices have been employed to detect motion errors in ski jumps.

In [23], the angle of attack, yaw angle, and edge angle were evaluated to determine a range for each angle to perfectly achieve the jump. In [24], a constant equivalent fall height jump was built and tested. Results showed that low values of equivalent fall height (EFH) are practical to build, and landing can be controlled through the design of the landing surface.

This is the only paper [25] among all the ski jump studies that uses simulations to visualize the jumps. However, it does not provide much information about the development of the jump model and related aspects.

B. PTWs jumps

Paper [8] highlighted dominant movement patterns during critical phases such as the start, takeoff, landing, and pumping, particularly focusing on hip and knee extension, as well as shoulder abduction and adduction. The authors suggest that further research is needed to explore power and acceleration profiles in BMX, as this information would help establish best practices for athlete training and preparation. While the experiments detailed in this paper yield valuable insights for evaluating jumps, they involve human participants performing the jumps, which poses inherent dangers.

In conclusion, most of my literature reviews focus on detecting ski jumps using either sensors or camera systems to collect data, which is then

evaluated to improve future jumps. There is only one paper that includes the use of simulation, but it does not provide much useful information. Regarding PTW (powered two-wheeler) jumps, there's only one paper being found discussing experimental data, but it does not involve any simulation and instead relies on physical experiments with human participants. This indicates a significant gap in the literature when it comes to the simulation of motorcycle or PTW jumps, and the purpose of my work here is to fill this gap.

III. Methodology

In this project, all the simulations will be conducted in a high-fidelity simulation software,Webots[27]. while the jump is modeled and implemented using JavaScript in the **P5.js** platform[28].

A. jump design

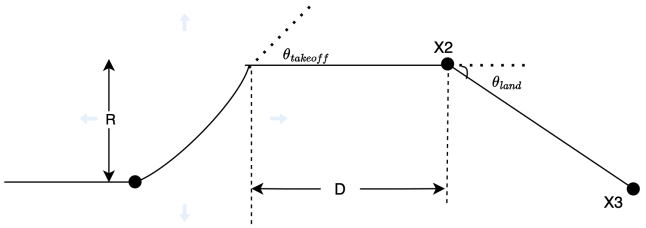

Fig 1. Jump parameters

Figure 1 shows the jump design includes all the key parameters. R represents the radius of the takeoff ramp radius in meters. D denotes the jump distance. The$\theta_{takeoff}$ refers the taking off angle and θ_{land} refers to the landing angle.

In this project, I will use of the the jump I designed early and the dimension is shown in Table1.

D(m)	R(m)	$\vartheta_t(rad)$	$\theta_l(rad)$
20	10	0.7	0.349

Table1. two different jump size

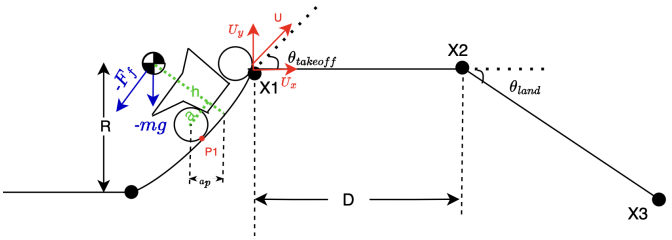

Fig 2. jump's parameters

a. jumping speed

As shown in Figure 2, the speed can be decomposed into its horizontal and vertical components, represented by Equations (1) and (2):

$$U_x = U \cdot cos\theta_t \tag{1}$$

$$U_y = U \cdot sin\theta_t \tag{2}$$

The Free-Fall Height during the jump can be expressed as Equation (3) where tf is the flying time and g is the gravitational acceleration, equal to 9.81 m/s²:

$$h = \frac{1}{2}g(\frac{t_f}{2})^2 = \frac{1}{2}g(\frac{D}{2Ucos\theta_t})^2 \tag{3}$$

The maximum Theoretical Height can be expressed in Equation (4):

$$h = \frac{(U_y)^2}{2g} = \frac{(Usin\theta_t)^2}{2g} \tag{4}$$

From Equations (3) and (4), the required speed to clear the jumps can be expressed as Equation (5) in terms of the jump distance D and the takeoff angle θ_t:

$$U = \sqrt[4]{\frac{2g^2D^2}{8cos\theta_t^2 sin\theta_t^2}} \tag{5}$$

b. Rider design

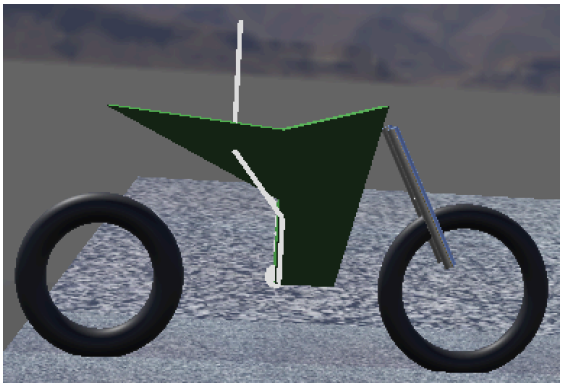

FIgure3. rider shown in Webots

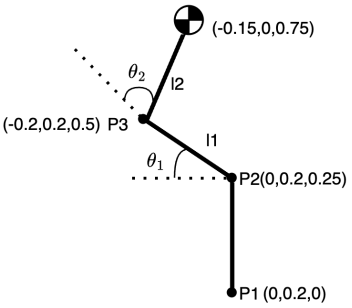

Figure 4. Rider's configuration in Webots

The Rider PROTO has been built in the Webots simulation using a JavaScript rod function to generate cylinders between the body points(ankle, knee, hip, torsi). It is designed as an add-on to the WeBike's PROTO. In the Rider PROTO, the rider's shin, thigh, and knee positions, as well as their dimensions and weight, are configurable. For this project, I measured my own body dimensions and used average weight values sourced online. The specific dimensions used for the rider in this project are provided in Table 3, and assumed the rider's CG is located in the middle of the torso:

total rider mass(kg)	70
thin length(m)	0.25
thigh length(m)	0.32
torso length(m)	0.255
thin weight(kg)	1.75
thigh weight(m)	5
torso weigh(m)	35

Table3. Rider's parameters

The inverse kinematic was used to determine where the thigh and toso is at by controlling the motor on the knee and hip. The rider's configuration is shown Figure 5:

In this figure, the shin is represented as the segment between Point P1 and P2, the thigh spans from Point P2 to P3, and the torso extends from Point P3 to the rider's center of gravity (CG). The ankle is positioned approximately at the location of the bike's footpeg, consistent with how a real rider sits.

In this project, I assume that the ankle remains stationary, allowing only the knee and hip joints to move. As a result, the shin segment stays fixed throughout the simulation. While this assumption is not entirely realistic, it significantly simplifies the model. Future work may incorporate ankle movement to more accurately simulate the rider's motion during a jump.

The joint angles θ_1 (knee) and θ_2 (hip) can be determined geometrically, as described in Equations 6 and 7. Additionally, the rider's center of gravity is assumed to move only in the x-direction.

$$\theta_1 = arctan(\frac{0.25}{0.2}) \quad (6)$$

$$\theta_2 = pi - \theta_1 - arctan(\frac{0.25}{0.05}) \quad (7)$$

So the CG of the rider's position(Px,Py) can be expressed in Equations below:

$$Px\_rest = l1 * np.cos(theta1\_rest) + l2 * np.cos(th$$

$$Py\_rest = l1 * np.sin(theta1\_rest) + l2 * np.sin(th$$

Based on these assumptions, inverse kinematics (IK) can be applied to compute the desired knee and

hip joint angles that will position the rider's center of gravity (CG) forward or backward by a specified distance. By solving the IK equations, I can determine the necessary motor commands to achieve the target CG shift along the x-axis.

c. Throttle torque

The Vehicle's parameters can be find in Table 1:

Table 1. Basic Vehicle Parameters

Parameter	Description	Value (bicycle)	Value (motorcycle)	Units
m_{ff}	Front frame mass	4	10	kg
m_{rf}	Rear frame mass	85	158	kg
m_{fw}	Front wheel mass	3	10	kg
m_{rw}	Rear wheel mass	3	13	kg
J_{yyf}	Front wheel spin M.O.I.	0.368	0.798	kg-m^2
J_{yyr}	Rear wheel spin M.O.I.	0.27	0.833	kg-m^2
x_{rf}	x-distance to rear frame C.G.	0.3	0.689	m
x_{ff}	x-distance to front frame C.G.	0.9	1.25	m
h_{rf}	Height of rear frame C.G.	0.9	0.519	m
h_{ff}	Height of front frame C.G.	0.7	0.735	m
R_{rw}	Radius of rear wheel	0.3	0.330	m
R_{fw}	Radius of front wheel	0.35	0.356	m
c	Trail (see Figure 1)	0.08	0.115	m
b	Wheelbase (see Figure 1)	1.02	1.45	m
λ	Rake angle (see Figure 1)	1.25	1.10	rad

Table 2. Vehicle's parameters

To determine the torque required to achieve the correct pitch angle at takeoff, moments are summed about the rear wheel contact point as shown in Equation (6). The sum of moments accounts for forces due to horizontal and vertical accelerations as well as the rider's forward acceleration. Assuming the bike and the rider is modeled as a rigid body:

$$\sum T_{p_1} = ma_x h + mga_p = J\ddot{\theta} \tag{6}$$

where J can be expressed as

$$J = mr^2 = m(a^2 + h^2),\ a_p = \frac{a}{cos\theta_t} - \frac{h}{sin\theta_t}$$

The bike's pitch rate can be expressed as Equation(7):

$$\dot{\theta} = -\frac{\theta_t + \theta_l}{t_f} \tag{7}$$

and pitch acceleration can be expressed as equation (8) where $\frac{U}{b}$ is bike's estimated takeoff time:

$$\ddot{\theta} = -\frac{(\theta_t + \theta_l)U}{t_f b} \tag{8}$$

Combining Equations (6), (7), and (8), the torque applied on the rear wheel during takeoff to achieve and maintain the correct pitch rate can be expressed as Equation (9):

$$\sum T_{p_1} = ma_x h + mga_p = m(a^2 + h^2)\frac{(\theta_t + \theta_l)U}{t_f b} \tag{9}$$

After simplification, the equation can be further expressed in a more compact form as Equation (10):

$$T_{takeoff} = \frac{(m(a^2 + h^2)\frac{(\theta_t+\theta_l)U}{t_f b} - mg(\frac{a}{cos\theta_t} - \frac{h}{sin\theta_t}))R_{rw}}{h} \tag{10}$$

B. FSM design

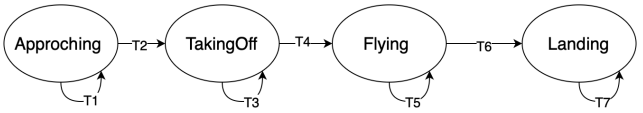

Fig.5 jump FSM design

For this motorcycle jump, four distinct states are involved. The first state is the Approaching state, which occurs before the bike's front wheel leaves the ground at point X1. The second state, Taking Off, begins when the front wheel lifts off the ground at X1 and continues until the rear wheel also leaves the ground. Once the rear wheel lifts off, the bike transitions into the Flying state, which lasts until the front wheel makes contact with the ground at point X2. The transition between each state depends on the bike's global x location obtained from GPS, under the guidance of the known jump parameters and locations.

C. Controller design

In this project, a proportional speed controller will be implemented for the Approaching state to regulate the bike's speed as it prepares for takeoff. During the Flying state, a PD controller will be used to control the pitch angle. By regulating the torques applied to the rear wheel, the controllers

can adjust the bike's pitch angle, ensuring alignment with the desired landing trajectory.

a. Speed Proportional controller

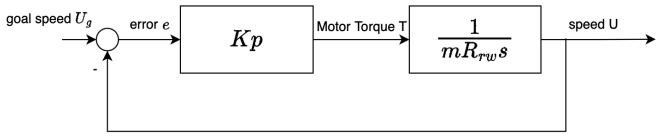

Fig.6 P speed control

Figure 4 illustrates the feedback Proportional Controller for the bike's speed. The input to the controller is the goal speed, derived from Equation (5), while the output is the bike's actual speed from GPS. The controller adjusts the torque applied to the rear wheel to minimize the speed error, ensuring the bike achieves the desired takeoff velocity for the jump. According to the Torque applied on the rear wheel as $T = m_b a_x R_{rw}$, so the plant transfer function is $P(s) = \frac{1}{mR_{rw}s}$

a. PD pitch angle control

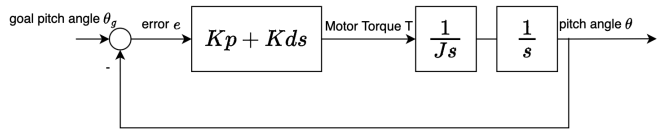

Fig7. PD pitch angle controller

Figure 6 shows the PD Pitch Angle Controller for the Flying state, where the goal pitch angle $\theta_g = -\theta_l$. The output of the system is the bike's actual pitch angle, measured by the IMU.

IV. Results

Figure 8 shows the comparison of the bike's pitch angle under an open-loop controller for different rider movements. The red line represents the case where the rider remains still, the black line shows the result when the rider moves forward, and the blue line indicates the rider moving backward.

In the simulation, the rider moves forward by 0.05 m. A larger movement was found to cause instability. In contrast, the rider moves backward by 0.1 m. Although the figure shows that moving backward results in a larger peak pitch angle during the jump, this cannot be directly compared to the forward case due to the difference in movement distances.

The key takeaway is that the bike always lands stably under open-loop control, regardless of the rider's motion. Additionally, the rider's forward or backward movement has minimal impact on the overall open-loop behavior in this simulation.

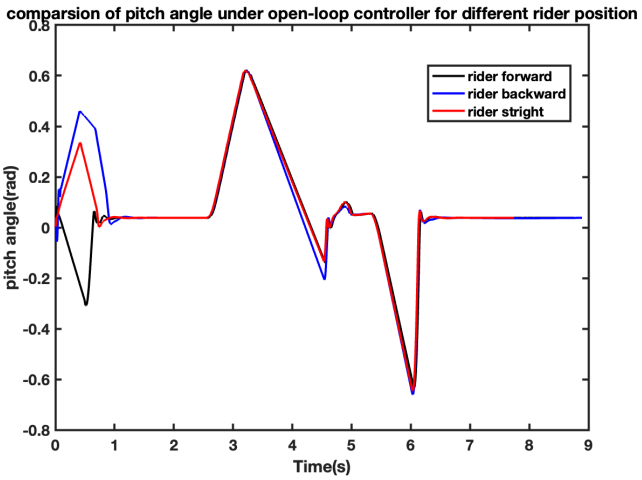

Fig 8. comparison of pitch angle under open loop controller

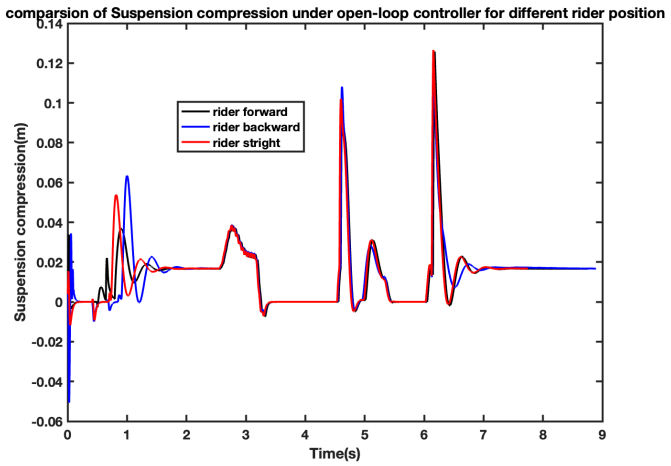

Figure 9. comparison of suspension compression under open loop controller

Figure 9 shows the comparison of suspension compression under the open-loop controller for different rider movements. Similar to the pitch angle results, all three cases exhibit very similar suspension behavior throughout most of the jump. However, when the rider moves backward, the simulation shows a slightly delayed landing compared to the other two cases.

This delay could come from that backward shift of the rider's center of gravity, which may slightly alter the bike's orientation and trajectory in the air. Also, the bike's rear frame becomes heavier and therefore a later ground contact, especially with the rear suspension.

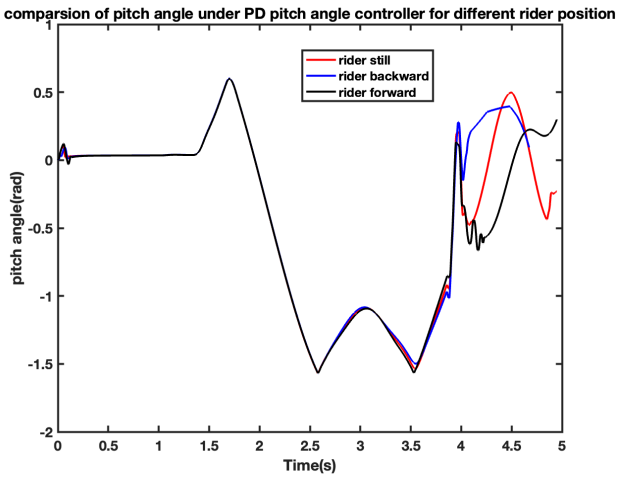

Figure 10. comparison of suspension compression under PD pitch angle controller

Figure 10 shows the bike's pitch angle during the jump under a PD pitch angle controller for different rider movements. As shown in the figure, the bike experiences a significantly larger pitch angle during the jump compared to the open-loop cases. This large oscillation likely explains why the system fails to stabilize at the end of the jump.

The cause could be related to inappropriate PD gains, which require further tuning and testing. Another possible reason is that the rider's movement increases the system's moment of inertia in pitch, making the dynamics more oscillatory under the current controller settings. Additionally, in this simulation, it's possible that the rear wheel is exceeding physical or simulation limits.

V. Conclusion and Future Work

In this project, I implemented a controller to control the rider's forward and backward motion during a motorcycle jump by applying inverse kinematics to control the knee and hip motors. All three rider movement scenarios were stable under an open-loop controller, the differences between them were minimal and not conclusive.

However, under the PD pitch angle controller, the system was unable to stabilize, and the bike experienced large pitch oscillations during the jump. This suggests that the current controller gains are not well-tuned and that the system's increased pitch inertia may be affecting stability.

In future work, I plan to complete and fine-tune the PD controller to determine optimal gain values, enabling a clearer comparison of the effects of rider movement. Additionally, I will implement a PI pitch rate controller for further testing and performance evaluation. I also intend to extend the rider model to include ankle movement, which may improve the accuracy of the rider dynamics and enhance control authority during the jump.

VII. References

[1] "Fatality Facts 2022: Motorcycles and ATVs," IIHS-HLDI crash testing and highway safety. Accessed: Nov. 13, 2024. [Online]. Available: https://www.iihs.org/topics/fatality-statistics/detail/motorcycles-and-atvs

[2] N. Petrone, M. Cognolato, J. A. McNeil, and M. Hubbard, "Designing, building, measuring, and testing a constant equivalent fall height terrain park jump," *Sports Eng.*, vol. 20, no. 4, pp. 283–292, Dec. 2017, doi: 10.1007/s12283-017-0253-y.

[3] N. Petrone, M. Cognolato, J. A. McNeil, and M. Hubbard, "Designing, building, measuring, and testing a constant equivalent fall height terrain park jump," *Sports Eng.*, vol. 20, no. 4, pp. 283–292, Dec. 2017, doi: 10.1007/s12283-017-0253-y.

[4] D. Levy, M. Hubbard, J. A. McNeil, and A. Swedberg, "A design rationale for safer terrain

park jumps that limit equivalent fall height," *Sports Eng.*, vol. 18, no. 4, pp. 227–239, Dec. 2015, doi: 10.1007/s12283-015-0182-6.

[5] M. Hubbard, J. A. McNeil, N. Petrone, and M. Cognolato, "Impact Performance of Standard Tabletop and Constant Equivalent Fall Height Snow Park Jumps", Accessed: Nov. 13, 2024. [Online]. Available: https://asmedigitalcollection.asme.org/astm-ebooks/book/2192/chapter/27892017/Impact-Performance-of-Standard-Tabletop-and

[6] M. Hubbard and A. Swedberg, "Design of Terrain Park Jump Landing Surfaces for Constant Equivalent Fall Height Is Robust to 'Uncontrollable' Factors," in *Skiing Trauma and Safety, 19th Volume*, ASTM International, pp. 1–20. doi: 10.1520/STP104515.

[7] A. Swedberg and M. Hubbard, "Modeling Terrain Park Jumps: Linear Tabletop Geometry May Not Limit Equivalent Fall Height," in *Skiing Trauma and Safety, 19th Volume*, ASTM International, pp. 1–16. doi: 10.1520/STP104335.

[8] J. F. Cowell, M. R. McGuigan, and J. B. Cronin, "Movement and Skill Analysis of Supercross Bicycle Motocross," *J. Strength Cond. Res.*, vol. 26, no. 6, p. 1688, Jun. 2012, doi: 10.1519/JSC.0b013e318234eb22.

[9] N. Sato, R. Takamagi, T. Takayama, and Y. Murata, "A Proposal of a Simplified Jumping Distance Measurement Method for Ski Jumper's Motion Monitoring System Using Terrestrial Magnetism and Acceleration Sensors," in *2012 26th International Conference on Advanced Information Networking and Applications Workshops*, Mar. 2012, pp. 1005–1010. doi: 10.1109/WAINA.2012.121.

[10] N. Sato, T. Takayama, and Y. Murata, "On Automatic Flying Distance Measurement on Ski Jumper's Motion Monitoring System," in *2012 15th International Conference on Network-Based Information Systems*, Sep. 2012, pp. 603–609. doi: 10.1109/NBiS.2012.50.

[11] N. Sato, T. Takayama, and Y. Murata, "Early Evaluation of Automatic Flying Distance Measurement on Ski Jumper's Motion Monitoring System," in *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, Mar. 2013, pp. 838–845. doi: 10.1109/AINA.2013.77.

[12] M. Oikawa, N. Sato, and Y. Murata, "A Proposal and Trial on a Model of a Motion Monitor System for a Ski Jumper Using Terrestrial Magnetism and Acceleration Sensors," in *2009 International Conference on Network-Based Information Systems*, Aug. 2009, pp. 364–369. doi: 10.1109/NBiS.2009.13.

[13] B. H. Groh, N. Weeger, F. Warschun, and B. M. Eskofier, "Simplified orientation determination in ski jumping using inertial sensor data," in *2014 DGON Inertial Sensors and Systems (ISS)*, Sep. 2014, pp. 1–11. doi: 10.1109/InertialSensors.2014.7049482.

[14] N. Sato, N. Umeki, Y. Murata, and A. Suzuki, "Instant Video Share and View System for Ski-Jump Training Using Cheap Devices," in *2016 19th International Conference on Network-Based Information Systems (NBiS)*, Sep. 2016, pp. 260–266. doi: 10.1109/NBiS.2016.17.

[15] L. Schulthess, T. M. Ingolfsson, M. Nölke, M. Magno, L. Benini, and C. Leitner, "Skilog: A Smart Sensor System for Performance Analysis and Biofeedback in Ski Jumping," in *2023 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, Oct. 2023, pp. 1–5. doi: 10.1109/BioCAS58349.2023.10389124.

[16] D. Štepec and D. Skočaj, "Video-Based Ski Jump Style Scoring from Pose Trajectory," in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, Jan. 2022, pp. 682–690. doi: 10.1109/WACVW54805.2022.00075.

[17] K. Ludwig, M. Einfalt, and R. Lienhart, "Robust Estimation of Flight Parameters for SKI Jumpers," in *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, Jul. 2020, pp. 1–6. doi: 10.1109/ICMEW46912.2020.9105973.

[18] B. H. Groh, J. Fritz, M. Deininger, H. Schwameder, and B. M. Eskofier, "Unobtrusive and wearable landing momentum estimation in Ski jumping with inertial-magnetic sensors," in *2018 IEEE 15th International Conference on*

Wearable and Implantable Body Sensor Networks (BSN), Mar. 2018, pp. 102–105. doi: 10.1109/BSN.2018.8329669.

[19]M. Bächlin, M. Kusserow, G. Tröster, and H. Gubelmann, “Ski jump analysis of an Olympic champion with wearable acceleration sensors,” in *International Symposium on Wearable Computers (ISWC) 2010*, Oct. 2010, pp. 1–2. doi: 10.1109/ISWC.2010.5665851.

[20]N. Islam, “An image processing based approach to analyse ski-jump’s length, height and velocity of an athlete,” in *2020 International Conference on Information Science and Communication Technology (ICISCT)*, Feb. 2020, pp. 1–5. doi: 10.1109/ICISCT49550.2020.9080054.

[21]C. L. Roberts-Thomson, A. M. Lokshin, and V. A. Kuzkin, “Jump detection using fuzzy logic,” in *2014 IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES)*, Dec. 2014, pp. 125–131. doi: 10.1109/CIES.2014.7011841.

[22]H. Brock and Y. Ohgi, “Assessing Motion Style Errors in Ski Jumping Using Inertial Sensor Devices,” *IEEE Sens. J.*, vol. 17, no. 12, pp. 3794–3804, Jun. 2017, doi: 10.1109/JSEN.2017.2699162.

[23]M. Virmavirta and J. Kivekäs, “Aerodynamics of an isolated ski jumping ski,” *Sports Eng.*, vol. 22, no. 1, p. 8, Feb. 2019, doi: 10.1007/s12283-019-0298-1.

[24]N. Petrone, M. Cognolato, J. A. McNeil, and M. Hubbard, “Designing, building, measuring, and testing a constant equivalent fall height terrain park jump,” *Sports Eng.*, vol. 20, no. 4, pp. 283–292, Dec. 2017, doi: 10.1007/s12283-017-0253-y.

[25]O. Michel, “Cyberbotics Ltd. Webots™: Professional Mobile Robot Simulation,” *Int. J. Adv. Robot. Syst.*, vol. 1, no. 1, p. 5, Mar. 2004, doi: 10.5772/5618.

[26]K. Marasovic, “Visualised interactive computer simulation of ski-jumping,” in *Proceedings of the 25th International Conference on Information Technology Interfaces, 2003. ITI 2003.*, Jun. 2003, pp. 613–618. doi: 10.1109/ITI.2003.1225411.

[27]Webots.http://www.cyberbotics.com .Open-source Mobile Robot Simulation Software [Accessed 15-Dec-2024]

[28]“p5.js.” Accessed: Dec. 17, 2024. [Online]. Available: https://p5js.org/