Fall Avoidance and Recovery for Bipedal Robots using Walking Sticks

Benjamin Tam, Navinda Kottege
Autonomous Systems Lab, CSIRO, Brisbane, QLD, Australia
{benjamin.tam, navinda.kottege}@csiro.au

Abstract

For humanoid robots to replace humans in dangerous environments, they require a robust fall prevention system that can detect falls, prevent major damage and recover back up to continue on with the task. This paper introduces a system that is capable of fall detection, mitigation and recovery in a robot equipped with walking sticks. Walking sticks allow the humanoid robot to extend its support polygon in a fall. Using IMU data and kinematics, the orientation of the robot is used to detect if external disturbances would cause a fall. The required position of the walking stick tips are calculated to minimise the angle fallen before the system is deployed. A recovery sequence is generated to recover the robot to a stable state prior to the disturbance. Hardware experiments on the Robotis OP2 found the system was capable of reducing the impact shock for forward falls while standing and walking.

1 Introduction

Legged robots, compared to wheeled robots have several features that make them ideal for traversing rough terrain and for disaster relief applications since they are able to step over obstacles and place their feet on small footholds while maintaining a level body [Todd, 1985].

Humanoid robots, such as the Robotis OP2 shown in Figure 1, are excellent candidates for deployment in man-made environments containing stairs, doors and controls designed for humans [Brooks *et al.*, 2004]. The DARPA Robotics Challenge (DRC) in 2015 used this scenario as motivation. The challenge highlighted the limited research in the field of fall detection, damage prevention and recovery as numerous robots fell and only one was undamaged enough to stand up.

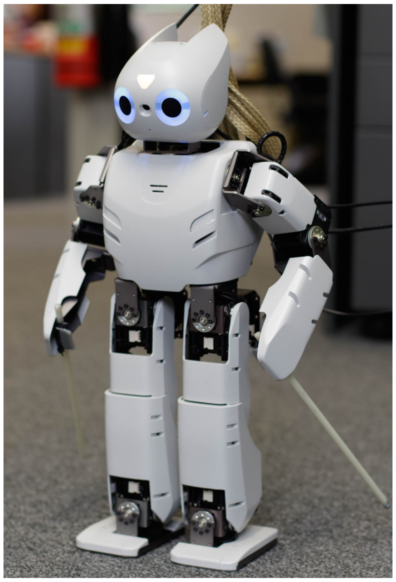


Figure 1: Robotis OP2 platform used for experiments.

Of the previous research conducted, different strategies have been proposed to prevent a robot damaging itself or its surroundings. Inspired by judo martial-arts techniques *ukemi*, [Fujiwara *et al.*, 2002; 2003; 2006] implemented a controller to limit damage by falling on specific shock-absorbing body parts. [Ha and Liu, 2015] extended these fall techniques to a generalised algorithm capable of multiple contact points planning for a variety of external disturbances to achieve zero final momentum. A tripod fall [Yun and Goswami, 2014] was designed to keep the centre of mass (CoM) relatively high to reduce the amount of potential energy (PE) converted to kinetic energy (KE) during a fall. This was achieved through using a swing leg and two arms as the three points of contact. [Yun *et al.*, 2009; Yun and Goswami, 2012; Goswami *et al.*, 2014] implemented a foot placement and inertia shaping controller to change the fall direction to prevent damage to objects around it. To detect instabilities in the robot, attitude sensors are used in [Renner and Behnke, 2006] to classify disturbances and perform preventative actions. A fall prevention system developed by [Park *et al.*, 2015] used an inertial-measurement unit (IMU) for detection and a forward step with a swing leg to stop the robot falling.

The use of walking sticks or canes is relatively new within humanoid robots, with previous works focusing on adapting the gait of the robot to utilise the extra degrees of freedom for traversing rough terrain. The ski-type walking approach [Wang *et al.*, 2014; Wang and Zheng, 2015] enlarged the support area and stability margin of the robot using two canes. They verified the optimal 'crawl' sequence and cane length that increased stability assuming the arms are unable to bear high loads. An adaptive algorithm was proposed by [Kobayashi *et al.*, 2014] to optimise stability and efficiency across different terrain using a walking cane. The cane was selected for preventative, cyclic or leg-like usage with each type increasing the load bearing of the cane. SupraPeds [Khatib and Chung, 2014] added vision and force sensors to actuated smart staffs for contact-supported locomotion in unstructured terrain. Simulation results showed the whole-body control framework capable of multi-contact locomotion in 3D unstructured terrain.

The rest of the paper is structured as follows: The system and its components are introduced in Section 2, Section 3 describes the experimental setup, with the results shown in Section 4. Section 5 discusses the results and the conclusion follows in Section 6.

2 Fall Avoidance and Recovery System

A humanoid robot experiences various motions in normal operation, as shown in Figure 2. Of particular interest are the *Unstable Standing* and *Unstable Walking* states which lead to *Falling*. Ideally, all falls should be *Safe Falls* which allows for recovery back to *Stable Standing*.

Used extensively in legged robots is the support polygon, defined as the convex hull of the contact points of a robot to the ground [Kajita and Espiau, 2008]. Humanoid robots are susceptible to falls because of the small support polygon formed by the feet. Inspired by multi-legged robots' ability to traverse rough terrain due to a large support polygon, bipedal robots can also enlarge this by increasing the number of contact points. Observations in humans show crawling as a solution, although this places excessive forces on the hand and knee joints and necessitates a range of motion that most bipedal platforms are not designed for. Another solution is walking aids, used by the elderly and mountaineers, allowing the body to remain vertical and increasing the region of contact point positions. Walking aids can be stowed during times the hands are required to complete tasks, then reattached when fall prevention is required. Using walking sticks for fall prevention allows the robot to continue walking in the more efficient bipedal mode without the increased risk of falls. The idea of walking aids is not new; however, the proposed system is the first to the authors' knowledge that describes using walking sticks for fall prevention, damage reduction and recovery.

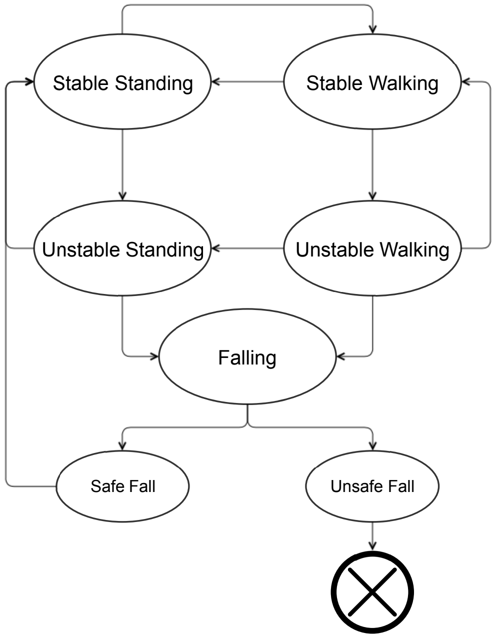


Figure 2: State diagram for the motion of a humanoid robot. Normal operation exists in the four stable and unstable states using a balance controller. A fall controller is required for large disturbances that cause a fall.

The Fall Avoidance and Recovery (FAR) system is comprised of different subsystems as outlined in Figure 3. The main components: *IMU Filter*, *Fall Classifier*, *Fall Controller* and *Fall Recovery* are described in detail in the following sections.

2.1 IMU Filter

The *IMU Filter* passes the raw IMU data into filters to provide orientation data for the *Fall Classifier*. Specially, six-axis IMU data (accelerometer and gyroscope) is passed through low-pass infinite impulse response (IIR) filters separately to reduce noise. The orientation is calculated from the noise-reduced accelerometer and gyroscope data independently. From the accelerometer, Equation 1 for pitch and 2 for roll are used to calculate the orientation. The gravity vector G_p is given by $[G_{px}\ G_{py}\ G_{pz}]^T$ when rotation occurs.

$$\tan(\theta_{xyz}) = \frac{-G_{px}}{\sqrt{{G_{py}}^2 + {G_{pz}}^2}} \qquad (1)$$

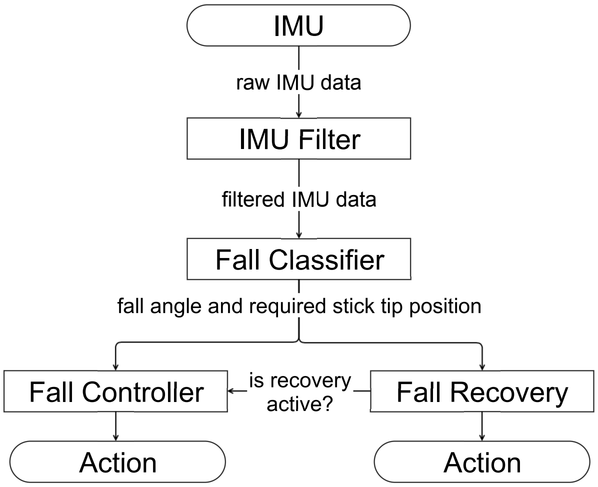


Figure 3: Overview of the FAR system.

$$\tan(\phi_{xyz}) = \frac{G_{py}}{\operatorname{sign}(G_{pz})\sqrt{{G_{pz}}^2 + {G_{px}}^2}} \tag{2}$$

The angular velocities provided by the gyroscope are integrated to get the orientation. A complementary filter (Equations 3 and 4) fuses both of these calculated orientation to provide a response that reacts quickly to disturbances (high pass filter of gyroscope) while rejecting the drift from the gyroscope (low pass filter of accelerometer) [Calton, 2007]. Only the pitch and roll angles are calculated as yaw is not required.

$$\theta = \alpha\ (\theta_p + \dot{\theta}_{gyro} \times dt) + (1 - \alpha)\ \theta_{xyz} \tag{3}$$

$$\phi = \alpha\ (\phi_p + \dot{\phi}_{gyro} \times dt) + (1 - \alpha)\ \phi_{xyz} \tag{4}$$

where α is the filter factor, $(\theta_p + \dot{\theta}_{gyro} \times dt)$ and $(\phi_p + \dot{\phi}_{gyro} \times dt)$ are the integration of the angular velocity from the gyroscope for orientation and θ_{xyz}/ϕ_{xyz} are from Equations 1 and 2 respectively. The *IMU Filter* outputs the orientation as filtered data for the rest of the system to use.

2.2 Fall Classifier

The *Fall Classifier* detects falls using the orientation from the *IMU Filter*. Humanoid robots have been modelled using various inverted pendulum models (IPM) [Orin *et al.*, 2013] to simplify stability and fall analysis. Previous models such as the linear IPM have assumed the height of the CoM is much greater than the length of the foot for the length of the foot to be negligible. This assumption does not hold for robots with large feet such as the Robotis OP2. Thus for a forward fall, the pivot moves to the front edge of the feet. This can be extended to backwards and sidewards falls using the appropriate edges of the feet.

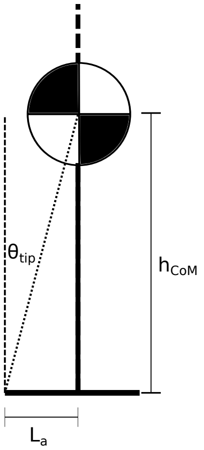


Figure 4: Inverted pendulum model (IPM).

Using the model, the CoM is located behind the unstable equilibrium point. The resultant equations of motion (EoM) have a restoration force caused by gravity when the CoM is behind the leading edge of the feet. This necessitates bounds on the orientation of the robot to ensure the equations remain valid.

The maximum angle the robot can tilt before it falls is equal to the initial angle between the CoM and the front edge of the feet given by Equation 5 and illustrated by Figure 4. When this angle (θ_{tip}) is passed, the CoM is no longer within the support polygon and a fall would occur. This model can be used for fall detection for static walking as the CoM is within the support polygon at all times.

$$\theta_{tip} = \frac{\pi}{2} - arctan\left(\frac{h_{CoM}}{L_a}\right) \tag{5}$$

In Equation 5, h_{CoM} is the height of the CoM and L_a is the distance from the front edge to the CoM in the horizontal direction.

External disturbances that cause a fall are predicted before θ_{tip} is reached by:

$$\omega_{tip}^2 = \omega_0^2 + \frac{2g}{h_{CoM}}\ (\cos(\theta_{tip}) - \cos(\theta_0)) \tag{6}$$

where θ_{tip} is the tip angle, θ_0 is the current θ, ω is the angular velocity ($\dot{\theta}$), ω_{tip} is ω at θ_{tip} and ω_0 is the current ω. Using the current orientation and angular velocity, the angular velocity at θ_{tip} can be calculated. If $\omega_{tip}^2 > 0$, the robot would continue to fall forwards when θ_{tip} is reached resulting in a fall.

Once a fall is detected, the trajectory of the falling CoM can be traced using the equation for an inverted pendulum. The time elapsed for the CoM to fall a certain angle is given in [Batista and Peternelj, 2006] as:

$$time = \sqrt{\frac{h_{CoM}}{2g}} \int_{\theta_0}^{\theta_{fall}} \frac{d\theta}{\sqrt{\frac{\omega_0^2 h_{CoM}}{2g} + \cos(\theta_0) - \cos(\theta)}} \tag{7}$$

where θ_{fall} is the fall angle. The fall angle is the amount the robot falls before the sticks can be moved to catch the robot from falling and is the angle at the final position of the deployed system. This is called the fallen deployed state. The variables that affect the deployment time of the sticks to the required location for a given θ_{fall} is given by:

$$time = \frac{\omega_{m_{max}}}{\theta_{joint}} + t_{reaction} \tag{8}$$

where $\omega_{m_{max}}$ is the maximum speed of the motors, θ_{joint} is the maximum angle the joints move and $t_{reaction}$ is the reaction time calculated from the controller's update rate. θ_{joint} is calculated through solving the inverse kinematics (IK) for the arms given a value for θ_{fall}.

Equating 7 and 8 gives Equation 9 which is solved to find the minimum θ_{fall}.

$$\frac{\omega_{max}}{\theta_{joint}} + t_{reaction} = A \int_{\theta_0}^{\theta_{fall}} \frac{d\theta}{\sqrt{B + \cos(\theta_0) - \cos(\theta)}} \tag{9}$$

where $A = \sqrt{\frac{h_{CoM}}{2g}}$ and $B = \frac{\omega_0^2 h_{CoM}}{2g}$.

Equation 9 is adapted to Algorithm 1 where θ_{fall} is increased until Equations 7 and 8 are equal. The trapezoidal rule is used for integration and θ_{fall} is checked to be inside the working range of the system.

Algorithm 1 Algorithm for fall classifier

```
$\omega_{tip} = detect(\theta_0, \omega_0)$;
**if** $\omega_{tip}^2 > 0$ **then**
    i = 0;
    **while** $f_1 < f_2$ **do**
        $\theta_i = \theta(i) \in [\theta_0, \pi/2]$;
        $f_1(i) = \frac{dtime}{2}[fall(\theta_i, \omega_i) + fall_{i-1}] + f_2(i-1)$;
        $f_2(i) = joint(\theta_i)$;
        $\theta_{fall} = \theta_i$;
        i++;
    **return** $\theta_{fall}$;
```

$detect(\theta_0, \omega_0)$ is Equation 6 with $f_1(i)$ Equation 7 and $f_2(i)$ Equation 8. The algorithm assumes that the time required to move the arms is initially greater than the time for the robot to fall, and that the gradient for Equation 7 is greater than Equation 8 such that an intersect exists. This is generally the case as $t_{reaction}$ is non-zero and constant. In the event of no intersect, the external disturbance is too large for the system to prevent a fall. The value of θ_i at which the functions intersect is θ_{fall}.

With θ_{fall} calculated, the location for the walking stick tips in the global coordinate frame is given by:

$$x_{sticks} = h_{CoM} \times \sin(\theta_{fall}) + x_{sf} \tag{10}$$

where x_{sf} is a hand-tuned safety factor to prevent the robot from tipping over after the walking sticks have deployed. A transformation is used to convert the tip position from the global frame to the 'fallen' robot's frame, given by:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = [Rot(z, \theta_{fall}) \cdot Trans(0, h_{CoM}, 0)]^{-1} \begin{bmatrix} x_{sticks} \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{11}$$

where $Rot(z, \theta)$ is the homogeneous transformation matrix for a rotation about the z-axis and $Trans(a, b, c)$ for a translation. The (x, y) location of the tip and θ_{fall} of the fallen robot are then sent to the *Fall Controller* and *Fall Recovery*.

2.3 Fall Controller

The *Fall Controller* receives the required tip positions from the *Fall Classifier* and calculates the required joint angles using inverse kinematics. An IK solver is required with the same degrees of freedom (DOF) as the number of joints in the plane of the fall. For a forward fall in robots that have two pitch joints, such as the Robotis OP2, the required two DOF IK solver is given by:

$$\theta_2 = atan2\left(\pm\sqrt{1 - \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2}}, \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2}\right) \tag{12}$$

$$\theta_1 = atan2(y, x) - atan2(k_2, k_1) \tag{13}$$

where $k_1 = l_1 + l_2\cos(\theta_2)$ and $k_2 = l_2\sin(\theta_2)$, and θ_1 and θ_2 are the shoulder and elbow pitch joints respectively. The degrees of freedom of the IK solver is dependent on the robot platform used and direction of the fall. The *Fall Controller* commands the joint motors to move to the required angles instantly.

2.4 Fall Recovery

A recovery sequence by the *Fall Recovery* is activated once the *Fall Controller* has deployed the walking sticks. The recovery sequence moves the robot back into a stable standing state where it can continue on with the task. The robot's orientation must remain constant before entering *Fall Recovery* to ensure the robot is stationary

Table 1: Start and end times for each stage of the recovery sequence across one and a half periods.

Start	End	Description
0	0.5	right leg move forwards
0.5	1.0	left leg move forwards
1.0	1.5	both legs stand up
1.25	1.25	move arms back

in the fallen deployed state before attempting recovery. The sequence is generated from sine waves across one and a half periods following the steps in Table 1.

The sine waves provide a smooth trajectory for the foot tip's six DOF IK solver. The sequence assumes that once both legs have moved individually, the feet are flat on the ground, regaining the support polygon with the CoM inside. The sequence starts in the fallen deployed state by moving each leg individually forward before standing up. The arms are moved back instantaneously while the robot is standing up to shift the CoM backwards while not dragging along the ground. This sequence allows the robot to continue moving forwards even after a fall, saving energy. While recovery is active, the *Fall Controller* is disabled until the sequence is completed.

2.5 Optimal Walking Stick Length

The performance of the FAR system is reliant on the length of the walking sticks. To select the optimal length (l_{sticks}) the performance of the system for the parameters in Table 3 was evaluated. Figure 5 outlines the optimal region for l_{sticks} for the initial case of $\theta_0 = \pi/180$ and $\omega_0 = 0$. The fall angle was incrementally increased for the *Fall Controller* to solve the required joint angles. The system can only deploy the sticks for falls within a range of θ_{fall}, which is bounded by 'Min Fall' and 'Max Fall' in Figure 5. The 'Joint Speed' comes from solving Equation 9 for the speed required, averaged with reaction time, to move the arms for a particular θ_{fall}. If the sticks are too long they would touch the ground before the robot has passed θ_{tip} or if the sticks are too short the robot would already be on the ground. Table 2 lists the factors that need to be considered when selecting the optimal stick length. The optimisation does not produce an optimal point but shows the region for choosing the stick length. An estimate of the 'Joint Speed' can be used to select the optimal stick length for the system using 'Joint Speed (Min Fall)' as the minimum speed required for a given stick length in Figure 5.

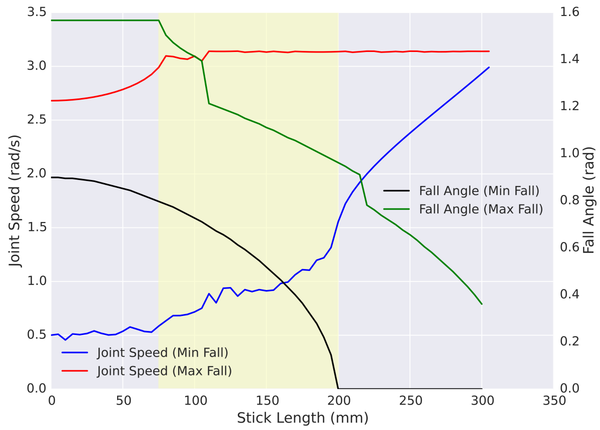


Figure 5: Performance of various walking stick lengths.

Table 2: Factors of selecting optimal walking stick length.

Factor	Description
θ_{fall}	Maximise the effectiveness of the system by ensuring the region of θ_{fall} contains most/all of the falls that external disturbances would cause
Joint speed	speed of motors and reaction time dependent on platform

3 Experiment Evaluation

3.1 Platform

The FAR system was implemented on the Robotis OP2 platform running the Robot Operating System (ROS) architecture. The Robotis OP2 shown in Figure 1 is a 20 DOF (2 x 6 DOF leg + 2 x 3 DOF arm + 2 DOF neck) 454.5 mm tall humanoid robot with a dual core Intel Atom processor. Implementing the system in ROS allows each component to be controlled and monitored individually.

Modifications were required to equip the robot arms with fibreglass walking sticks. 3D printed plastic holders were attached, shown in Figure 6, to allow the length of the walking sticks to be varied and fixed with screws.

The parameters used for the FAR system on the Robotis OP2 platform is listed in Table 3.

3.2 Experimental Setup

To test the FAR system, the robot was subjected to significant external disturbances. A pendulum test rig shown in Figure 7 was constructed to ensure repeatable disturbances in evaluating the system's performance. The pendulum's release point was preset so the contact

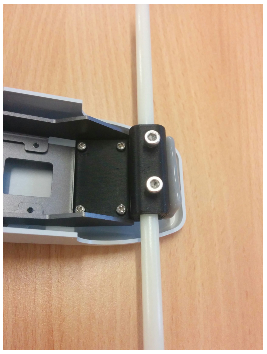


Figure 6: Walking stick holders.

Table 3: Parameters of the hardware platform.

Parameter	Value
h_{CoM}	0.318 m
L_a	0.052 m
θ_{tip}	9.29°
α	0.96
ω_{max}	5.0 rad/s
$t_{reaction}$	0.15 s
x_{sf}	0.100 m
l_{sticks}	0.150 m

force remains constant. Two pendulum release points were tested with a steel cap safety boot as the pendulum mass. Table 4 lists the specifications of the rig.

Table 4: Specification of experimental setup.

Specification	Value
Mass of pendulum	0.717 kg
Length of pendulum	0.930 m
Height of pendulum pivot	1.300 m
Position 1 (P1) release angle	17.55°
Position 2 (P2) release angle	26.90°
Δ height$_{P1}$	0.042 m
Δ height$_{P2}$	0.098 m

Three experiments were conducted to evaluate the performance of the FAR system. With the pendulum at the lower position (P1), the system was tested while the robot was standing. For the higher position (P2), both standing and walking were tested. These experiments were repeated without the FAR system for comparison.

Impact shock is the performance criteria used to evaluate the system. It is a suitable candidate as rapid deceleration can damage components. The internal IMU is used to record the change in acceleration experienced during the fall.

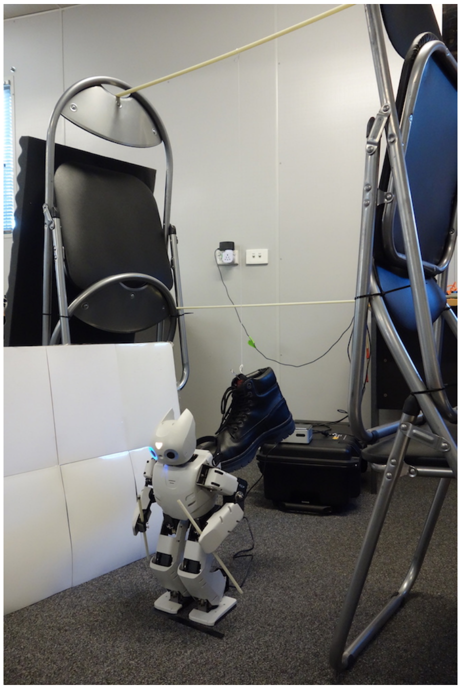


Figure 7: Experimental setup.

4 Results

The three experiments tested the system across different scenarios[1]. Table 5 lists the results for the different configurations. Using data collected during the Position 2 standing tests, the reaction time of the FAR system was calculated. These times are presented in Table 6.

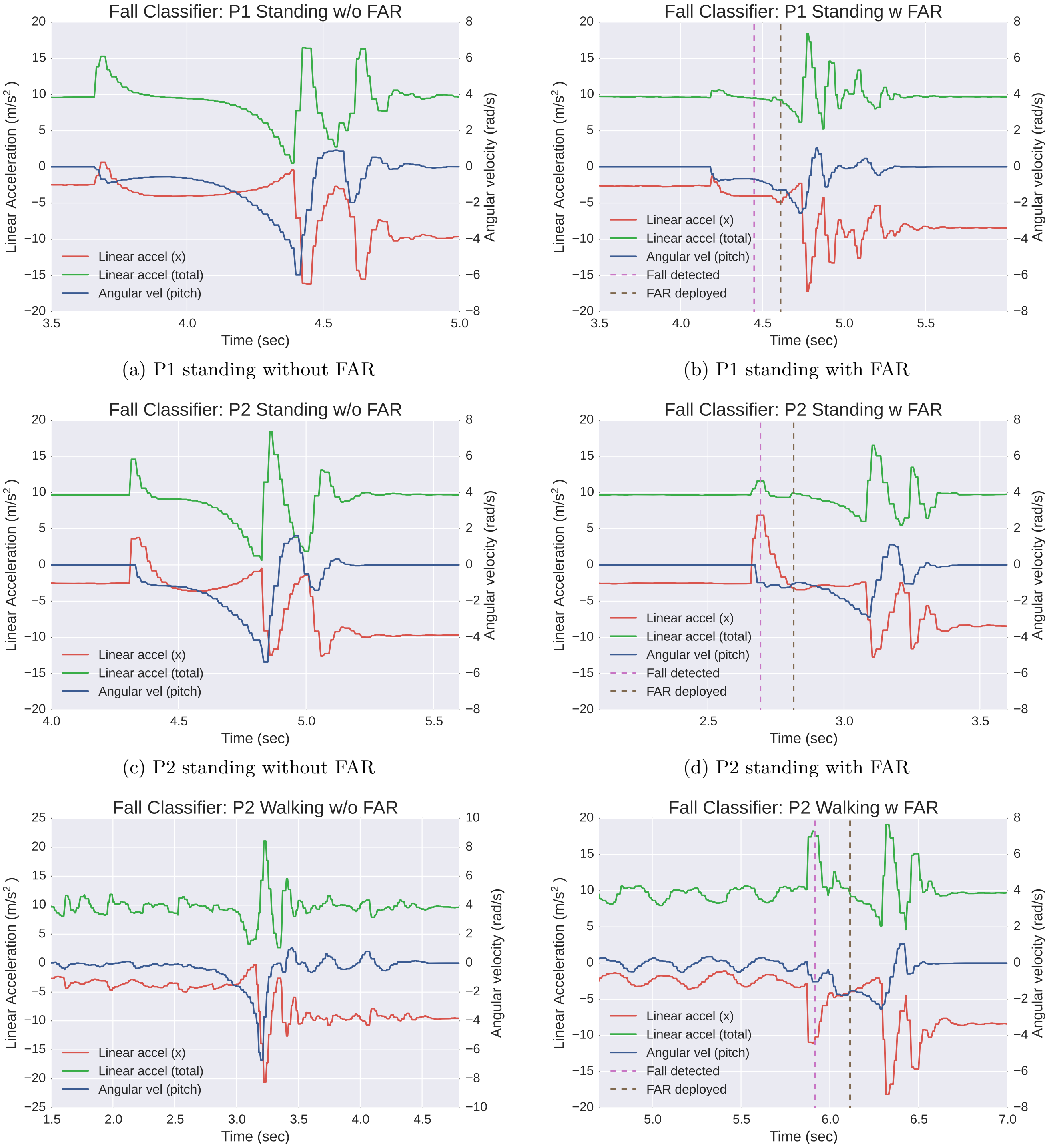
Table 5: Results of impact shock (g-force).

	Without FAR	With FAR	Decrease
P1 Standing	1.619 ± 0.045	1.347 ± 0.039	16.8%
P2 Standing	1.710 ± 0.065	1.357 ± 0.035	20.6%
P2 Walking	1.803 ± 0.107	1.320 ± 0.042	26.8%

Table 6: Reaction time of system.

	Reaction time
Fall detected	24 ± 5 ms
FAR deployed	132 ± 26 ms

[1] Please visit https://research.csiro.au/robotics/bipedal/ for video showing experimental results.



(a) P1 standing without FAR

(b) P1 standing with FAR

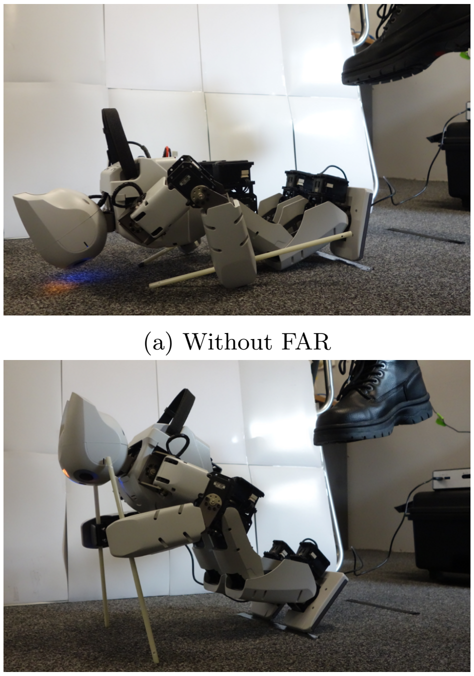
(c) P2 standing without FAR

(d) P2 standing with FAR

(e) P2 walking without FAR

(f) P2 walking with FAR

Figure 8: Data collected comparing the acceleration experienced by the robot without (a,c,e) and with (b,d,f) the FAR system. The maxima and minima of the total acceleration (green) are analysed for the impact shocks. The negative x-direction of linear acceleration points forwards from the robot's body which is not parallel to the ground when stationary.



(a) Without FAR

(b) With FAR

Figure 9: Final position after external disturbance. In (b), the *Fall Controller* has deployed the sticks but before *Fall Recovery* is activated. This is the fallen deployed state (θ_{fall}).

5 Discussion

The results demonstrate the system reduces the impact of a fall. In constantly calculating the angular velocity at θ_{fall}, the robot maximises the time it has to deploy the sticks to the required position. Observations of the *Fall Classifier* detecting a fall using only θ_{fall} as a threshold value showed the system would fall to the ground before the walking sticks could start to deploy. This shows that the ability to detect a fall before it happens increases the chances of mitigating the fall.

The walking sticks decreased the falling distance of the robot, with the CoM with FAR 200 mm higher than without. This is shown in Figure 9. The results of lower impact shocks supports the theory of the tripod fall in [Yun and Goswami, 2014] where the authors suggest conserving the PE reduces the KE at impact. The fallen deployed state is a new configuration that is able to reduce the KE of the system through an elevated CoM.

IMU data shows the robot without FAR experiences free-fall with the total linear acceleration near zero as shown in Figure 8a and 8c. Although the peak accelerations with FAR seems to be higher, such as comparing Figure 8a and 8b, this is due to the IMU measuring the gravity reaction force on the the robot which has a greater effect on with FAR than without.

The selection of the optimal walking stick length varies from previous works, most notably [Wang *et al.*, 2014], due to the different variables that affect its selection. While a longer length is advantageous for enhanced locomotion, it could hinder the deployment of the sticks in fall prevention. This relationship, shown in Figure 5, highlights the fact that shorter sticks gives the system longer time to respond. However, a longer length is ideal to reduce θ_{fall}, provided system deployment is not impeded.

5.1 Limitations

The *Fall Controller* does not consider the trajectory of the walking sticks when it is deployed. Thus, collision with the ground before θ_{fall} may occur. This would cause the robot to topple forwards along the sticks. To reduce this possibility, the stick length should be shortened.

Fall Recovery is currently unable to consistently recover from external disturbances where θ_{fall} is too large. Experiments on the Robotis OP2 showed the recovery sequence is limited by the ability to move the legs so that the feet are beneath the CoM. This is due to the self-collision of the legs and the body at the hip joint. To overcome the current limitation of the sequence, an additional step is required to either: use the sticks to push the upper body backwards to be upright for the hip joint to increase its range, or to drop the robot on its knees and push with the sticks to rock the robot backwards onto its feet.

6 Conclusion and Future Work

This work proposed a fall prevention system that utilised walking sticks to reduce the impact of falls. The *Fall Classifier* detects external disturbances that would cause a fall before the tip angle is reached, and the *Fall Controller* moves the walking sticks the shortest distance to stop the robot falling. A recovery sequence by the *Fall Recovery* returns the robot to a stable standing position. The proposed system was tested on the Robotis OP2 and showed successful results of the system in reducing the impact shock.

In future work, a mass-spring-damper model will be implemented on the arms and the system will be modified for directional falls. For directional falls, an additional *Fall Direction* component before the *Fall Classifier* is required to detect the fall direction along the transverse plane and select the most appropriate parameters for the IPM.

Acknowledgements

This work was supported in part by CSIRO's I3Hub project. The authors would like to thank Marko

Bjelonic, Michael Thoreau and Dylan Williamson for their support during the project.

References

[Batista and Peternelj, 2006] M. Batista and J. Peternelj. The Falling Time of an Inverted Plane Pendulum. *ArXiv Physics e-prints*, July 2006.

[Brooks *et al.*, 2004] R. Brooks, L. Aryananda, A. Edsinger, P. Fitzpatrick, C. C. Kemp, U. M. O'Reilly, E. Torres-Jara, P. Varshavskaya, and J. Weber. Sensing and manipulating built-for-human environments. *International Journal of Humanoid Robotics*, 1(01):1–28, 2004.

[Calton, 2007] S. Calton. The balance filter: A simple solution for integrating accelerometer and gyroscope measurements for a balancing platform. PowerPoint, 2007.

[Fujiwara *et al.*, 2002] K. Fujiwara, F. Kanehiro, S. Kajita, K. Kaneko, K. Yokoi, and H. Hirukawa. Ukemi: falling motion control to minimize damage to biped humanoid robot. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, pages 2521–2526 vol.3, 2002.

[Fujiwara *et al.*, 2003] K. Fujiwara, F. Kanehiro, S. Kajita, K. Yokoi, H. Saito, K. Harada, K. Kaneko, and H. Hirukawa. The first human-size humanoid that can fall over safely and stand-up again. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pages 1920–1926 vol.2, Oct 2003.

[Fujiwara *et al.*, 2006] K. Fujiwara, S. Kajita, K. Harada, K. Kaneko, M. Morisawa, F. Kanehiro, S. Nakaoka, and H. Hirukawa. Towards an optimal falling motion for a humanoid robot. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 524–529, Dec 2006.

[Goswami *et al.*, 2014] A. Goswami, S.K. Yun, U. Nagarajan, S. H. Lee, K. K. Yin, and S. Kalyanakrishnan. Direction-changing fall control of humanoid robots: theory and experiments. *Autonomous Robots*, 36(3):199–223, 2014.

[Ha and Liu, 2015] S. Ha and C. K. Liu. Multiple Contact Planning for Minimizing Damage of Humanoid Falls. *IEEE IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.

[Kajita and Espiau, 2008] S. Kajita and B. Espiau. *Springer Handbook of Robotics*, chapter Legged Robots, pages 361–389. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[Khatib and Chung, 2014] O. Khatib and S.Y. Chung. SupraPeds: Humanoid contact-supported locomotion for 3d unstructured environments. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2319–2325, May 2014.

[Kobayashi *et al.*, 2014] T. Kobayashi, K. Sekiyama, T. Aoyama, Y. Hasegawa, and T. Fukuda. Optimal selection of cane usage with humanoid robot. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 199–204, Nov 2014.

[Orin *et al.*, 2013] D. Orin, A. Goswami, and S.H. Lee. *Autonomous Robots*, chapter Centroidal Dynamics of a Humanoid Robot, pages 161–176. Springer, 2013.

[Park *et al.*, 2015] G. M. Park, S. H. Baek, and J. H. Kim. *Robot Intelligence Technology and Applications 3: Results from the 3rd International Conference on Robot Intelligence Technology and Applications*, chapter Falling Prevention System from External Disturbances for Humanoid Robots, pages 97–105. Springer International Publishing, Cham, 2015.

[Renner and Behnke, 2006] R. Renner and S. Behnke. Instability detection and fall avoidance for a humanoid using attitude sensors and reflexes. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2967–2973, Oct 2006.

[Todd, 1985] D. J. Todd. *A brief history of walking machines*, pages 11–33. Springer US, Boston, MA, 1985.

[Wang and Zheng, 2015] H. Wang and Y. F. Zheng. Torque and contact force analysis of ski-type walking for humanoid robots. In *2015 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, pages 1–6, May 2015.

[Wang *et al.*, 2014] H. Wang, S. Li, Y.F. Zheng, T. Kim, and P. Oh. Ski-type self-balance biped walking for rough terrain. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 3452–3457, May 2014.

[Yun and Goswami, 2012] S. K. Yun and A. Goswami. Hardware experiments of humanoid robot safe fall using aldebaran nao. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 71–78, May 2012.

[Yun and Goswami, 2014] S. K. Yun and A. Goswami. Tripod fall: Concept and experiments of a novel approach to humanoid robot fall damage reduction. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2799–2805, May 2014.

[Yun *et al.*, 2009] S. K. Yun, A. Goswami, and Y. Sakagami. Safe fall: Humanoid robot fall direction change through intelligent stepping and inertia shaping. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 781–787, May 2009.