UNIVERSITÀ DEGLI STUDI
DI GENOVA

ISTITUTO ITALIANO
DI TECNOLOGIA

PhD Program in Bioengineering and Robotics

Fall Prediction and Controlled Fall for Humanoid Robots

by

Rajesh Subburaman

Thesis submitted for the degree of *Doctor of Philosophy* (31st cycle)

December 2018

Prof. Nikos Tsagarakis	Main Supervisor
Dr. Jinoh Lee	Co-Supervisor

Thesis Jury:

Prof. Michael Mistry, *University of Edinburgh*	External examiner
Dr. Alexander Sproewitz, *Max Planck Research Group*	External examiner

Dibris

Department of Informatics, Bioengineering, Robotics and Systems Engineering

Abstract

Humanoids which resemble humans in their body structure and degrees of freedom are anticipated to work like them within infrastructures and environments constructed for humans. In such scenarios, even humans who have exceptional manipulation, balancing, and locomotion skills are vulnerable to fall, humanoids being their approximate imitators are no exception to this. Furthermore, their high center of gravity position in relation to their small support polygon makes them more prone to fall, unlike other robots such as quadrupeds. The consequences of these falls are so devastating that it can instantly annihilate both the robot and its surroundings. This has become one of the major stumbling blocks which humanoids have to overcome to operate in real environments. As a result, in this thesis, we have strived to address the imminent fall over of humanoids by developing different control techniques. The fall over problem as such can be divided into three sub-issues: fall prediction, controlled fall, and its recovery. In the presented work, the first two issues have been addressed, and they are presented in three parts.

First, we define what is fall over for humanoids, different sources for it to happen, the effect fall over has both on the robot and to its surroundings, and how to deal with them. Following which, we give a brief introduction to the overall system which includes both the hardware and software components which have been used throughout the work for varied purposes.

Second, the first sub-issue is addressed by proposing a generic method to predict the falling over of humanoid robots in a reliable, robust, and agile manner across various terrains, and also amidst arbitrary disturbances. The aforementioned characteristics are strived to attain by proposing a prediction principle inspired by the human balance sensory systems. Accordingly, the fusion of multiple sensors such as inertial measurement unit and gyroscope (IMU), foot pressure sensor (FPS), joint encoders, and stereo vision sensor, which are equivalent to the human's vestibular, proprioception, and vision systems are considered. We first define a set of feature-based fall indicator variables (FIVs) from the different sensors, and the thresholds for those FIVs are extracted analytically for four major disturbance scenarios. Further, an online threshold interpolation technique and an impulse adaptive counter limit are proposed to manage more generic disturbances. For the generalized prediction process, both the instantaneous and cumulative sum of each FIVs are normalized, and a suitable value is set as the critical limit to predict the fall over. To determine the best combination and the usefulness of multiple sensors, the prediction performance is evaluated on four different types of terrains, in three unique combinations: first, each feature individually with their respective FIVs; second, an intuitive performance based (PF); and finally, Kalman filter based (KF) techniques, which involve the usage of multiple features. For PF and KF techniques, prediction performance evaluations are carried out with and without adding noise. Overall, it is reported that KF performs better than PF and individual sensor features under different conditions. Also, the method's ability to predict fall overs during the robot's simple dynamic motion is also tested and verified through simulations. Experimental verification of the proposed prediction method on flat and uneven terrains was carried out with the WALK-MAN humanoid robot.

Finally, in reference to the second sub-issue, i.e., the controlled fall, we propose two novel fall control techniques based on energy concepts, which can be applied online to mitigate the impact forces incurred during the falling over of humanoids. Both the techniques are inspired by the break-fall motions, in particular, Ukemi motion practiced by martial arts people. The first technique reduces the total energy using a nonlinear control tool, called energy shaping (ES) and further distributes the reduced energy over multiple contacts by means of energy distribution polygons (EDP). We also include an effective orientation control to safeguard the end-effectors in the event of ground impacts. The performance of the proposed method is numerically evaluated by dynamic simulations under the sudden falling over scenario of the humanoid robot for both lateral and sagittal falls. The effectiveness of the proposed ES and EDP concepts are verified by diverse comparative simulations regarding total energy, distribution, and impact forces.

Following the first technique, we proposed another controller to generate an online rolling over motion based on the hypothesis that multi-contact motions can reduce the impact forces even further. To generate efficient rolling motion, critical parameters are defined by the insights drawn from a study on rolling, which are contact positions and attack angles. In addition, energy-injection velocity is proposed as an auxiliary rolling parameter to ensure sequential multiple contacts in rolling. An online rolling controller is synthesized to compute the optimal values of the rolling parameters. The first two parameters are to construct a polyhedron, by selecting suitable contacts around the humanoid's body. This polyhedron distributes the energy gradually across multiple contacts, thus called energy distribution polyhedron. The last parameter is to inject some additional energy into the system during the fall, to overcome energy drought and tip over successive contacts. The proposed controller, incorporated with energy injection, minimization, and distribution techniques result in a rolling like motion and significantly reduces the impact forces, and it is verified in numerical experiments with a segmented planar robot and a full humanoid model.

Acknowledgements

I would like to thank my supervisors Dr.Jinoh Lee and Dr.Nikos Tsagarakis for taking time from their busy schedule to guide me during the entire course of my research work. I would also like to thank them for giving me the freedom to choose/define the necessary research methods during the pursuit of the research goal. Having said that, they were also kind enough to regularly monitor the status of my research work and actively criticize it when necessary, to ensure the progress of my work. I am also grateful for their valuable suggestions and guidance bestowed upon me during the drafting of several conference and journal papers, technical presentations, and also during the writing of my thesis. All the aforementioned contributions from my supervisors have immensely improved my adept at doing any kind of research activity, writing scientific papers and reviewing them, analyzing the data, etc.

Thanks to my lab technicians Luca Muratore, Giuseppe, Lorenzo Baccelliere, Paolo, Alessio Margan, Phil, Diego, and Marco for their invaluable support offered in the past 3 years. Luca and Giuseppe have been very helpful in software related issues both on the simulator and real hardware platforms. Special thanks to Luca for his support on the hardware platforms even during outside working hours. Lorenzo, Paolo, and Diego have made my life easier in fixing the hardware related issues, whenever I stumbled upon some during my experiments. With the electronic circuits, it has always been a nightmare, and I could defy them with timely assistance from Phil, Alessio, and Marco.

I cannot forget numerous interesting, intellectual conversations which I have had over the past three years with my fellow researchers and Ph.D. students in the lab. Those conversations have either directly or indirectly influenced my research work and I am grateful to them for that. Few notable ones to whom I am grateful are: Brian for helping me with machine learning in the early stages, Domingo for his advice on ubuntu and machine learning related issues, Dr. Yangwei You and Songyan Xin for sharing their expertise on simulator and control related topics, Dr. Dimitrios Kanoulas for his expert advice on the usage of vision systems, Prof. Roy Featherstone for his professional insights on impacts in general, Dr. John and Vishnu Dev for sharing their knowledge about actuator dynamics, Dr. Leonel Rozo for his guidance on Gaussian process-based learning and Dr. Joao for discussions on the same topic, Malgorzata and Arturo for their help and suggestions on XBotCore and hardware control related issues, Vignesh for helping me with the optitrack system, and Dr. ChengXu for the discussions on kinematics based control.

Last but not least, I would like to thank my parents, brother, uncle, and all my friends for their unconditional support and affection which they have showered on me during this endeavor. Mere thanking wouldn't be enough to show my gratitude to them. In the past three years I have had many setbacks and on a number of occasions I have fallen over and hit the deck with high energy, but the aforementioned people have always been there as a resilient platform to make sure I spring back up with the same amount of energy. Even though most of them didn't have an in-depth idea about my work, their fascination with robotics and the fact that I do my Ph.D. in that field have made them proud of me. This has always infused in me some extra energy to pursue my research work more vigorously.

Table of Contents

List of Figures

List of Tables

Part I

Introduction

Chapter 1

Humanoid Fall Overs

Humanoids, which resemble humans, are designed to function like them and are expected to operate autonomously either individually or along side humans in man-made environments and disaster sites. In such unstructured scenarios, falling over is inevitable even for humans who possess exceptional sensing capabilities and reflexes, humanoids are no exception to this. Further, compared to other multi-legged robots such as quadruped or hexapod robots, the humanoid's center of gravity (CoG) location is relatively high and the convex hull size of its feet are small, making it more vulnerable to fall over. A number of strategies had been proposed over the years to prevent the robot from falling over but their success is limited, making the fall over problem unavoidable and yet to be resolved. This issue has become one of the major stumbling blocks for the humanoids to operate in real environments, calling for more attention to resolve it.

1-1 What is a Fall Over?

To resolve the issue of humanoids falling over, it is paramount to first define and understand what is a fall over? There is a wide spread misconception that a humanoid's unstableness is equivalent to its fall over. While a falling over humanoid can be unstable, the vice-versa, i.e., an unstable humanoid does not necessarily lead to its fall over. This can be visualized with the state space diagram shown in Fig. 1-1 which has the following three regions of interest:

- $S_{stable} \in S$ - is a region formed by a set of states, in which, if the robot starts from one of the states it will continue to remain within the same region for all time period.
- $S_{unstable} \in S$ - set of all states where the robot is unstable but they can be brought inside S_{stable} through active control.

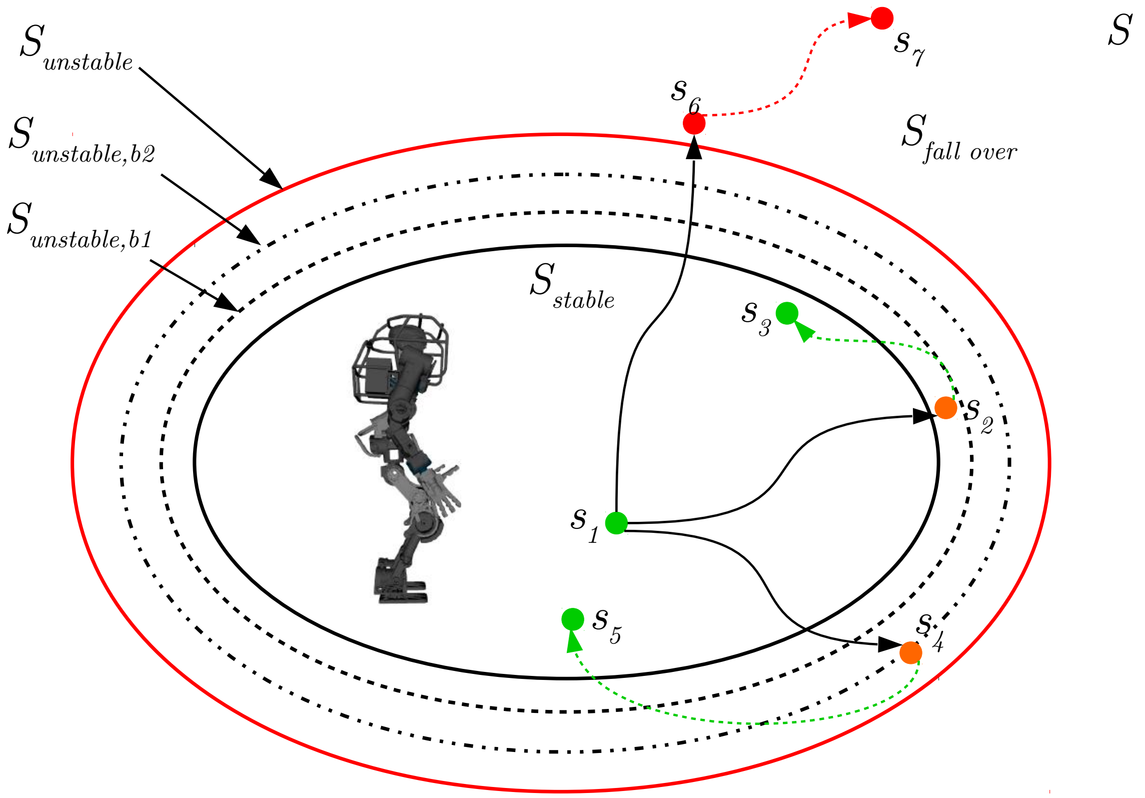


Figure 1-1: The complete state space of humanoids in general is represented by S with three regions of interest: stable (S_{stable}), unstable ($S_{unstable}$), and fall over ($S_{fallover}$). $S_{unstable,b1}$ and $S_{unstable,b2}$ denote the boundary of different balancing strategies and $s_\bullet$ is the state of the robot.

- $S_{fallover} \in S$ - set of all states which cannot be brought inside S_{stable} and continues to drift away until it is brought to rest by means of physical contact with the ground or any other fixed structure.

In Fig. 1-1, $S_{unstable,b1}$ and $S_{unstable,b2}$ represents the boundary of different balancing strategies, i.e., the set of unstable states which a particular control strategy can bring inside S_{stable} region, and this varies depending upon the effectiveness of each strategy. For instance, a simple CoG stabilization strategy can handle relatively less number of states when compared to a strategy based on angular momentum which makes use of the whole body of the robot. These strategies together expand the boundary of $S_{unstable}$. For example, if the robot starts at a state s_1 and due to some external disturbances ends up in s_2 or s_4, they can be brought inside S_{stable} by any balancing strategies whose boundaries are $S_{unstable,b1}$ and $S_{unstable,b2}$ respectively. However, if the external disturbance puts the robot at s_6, which is outside $S_{unstable}$, then the state continues to drift away from S_{stable} and comes to rest when it either hits the ground or some fixed structure. Hence the fall over state can be defined as follows:

A fall over state is the set of all states of a robot which cannot be driven back to the stable region (S_{stable}) by any means of active control and it continues to evolve until it makes contact with the ground or any fixed structure.

Mathematically the fall over state can be written as,

$$S_{\text{fall over}} = S - (S_{stable} \; \cup \; S_{unstable}). \tag{1-1}$$

1-2 Humanoid Fall Over Sources and Their Effects

The humanoid fall overs in reality can be instigated in numerous ways and they can be categorized depending upon the source as follows:

1. *software bugs*: fall overs due to either unintended control actions or inaction of a particular controller during a specific situation which has not been accounted for.

2. *operator errors*: unwise control actions commanded by the operator due to insufficient feedback of the robot's state or due to the operator's lack of experience with the system handling a particular task.

3. *real environments*: inability of the controller to adapt to the stochastic nature of real environments.

4. *external disturbances*: arbitrary impulses exerted on the robot in random directions either by humans, other robots, moving objects or terrain, etc.

5. *beta controllers*: controllers developed and tested in simulators and applied for the first time on real systems are more prone to generate erratic control actions which can result in fall overs.

6. *hardware failures*: failure of a particular sensor such as force/torque sensors, physical damage of a link or a joint actuator, etc., could hamper the planned control actions resulting in a robot's fall.

7. *communication failures*: delay in the communication of high level commands to the robot's on-board system or total loss of internal communications between the on-board system and actuators in critical situations could increase the chances of fall over.

8. *power cut-off*: cutting off the main power supply to the system will put the complete robot in passive mode resulting in its collapse.

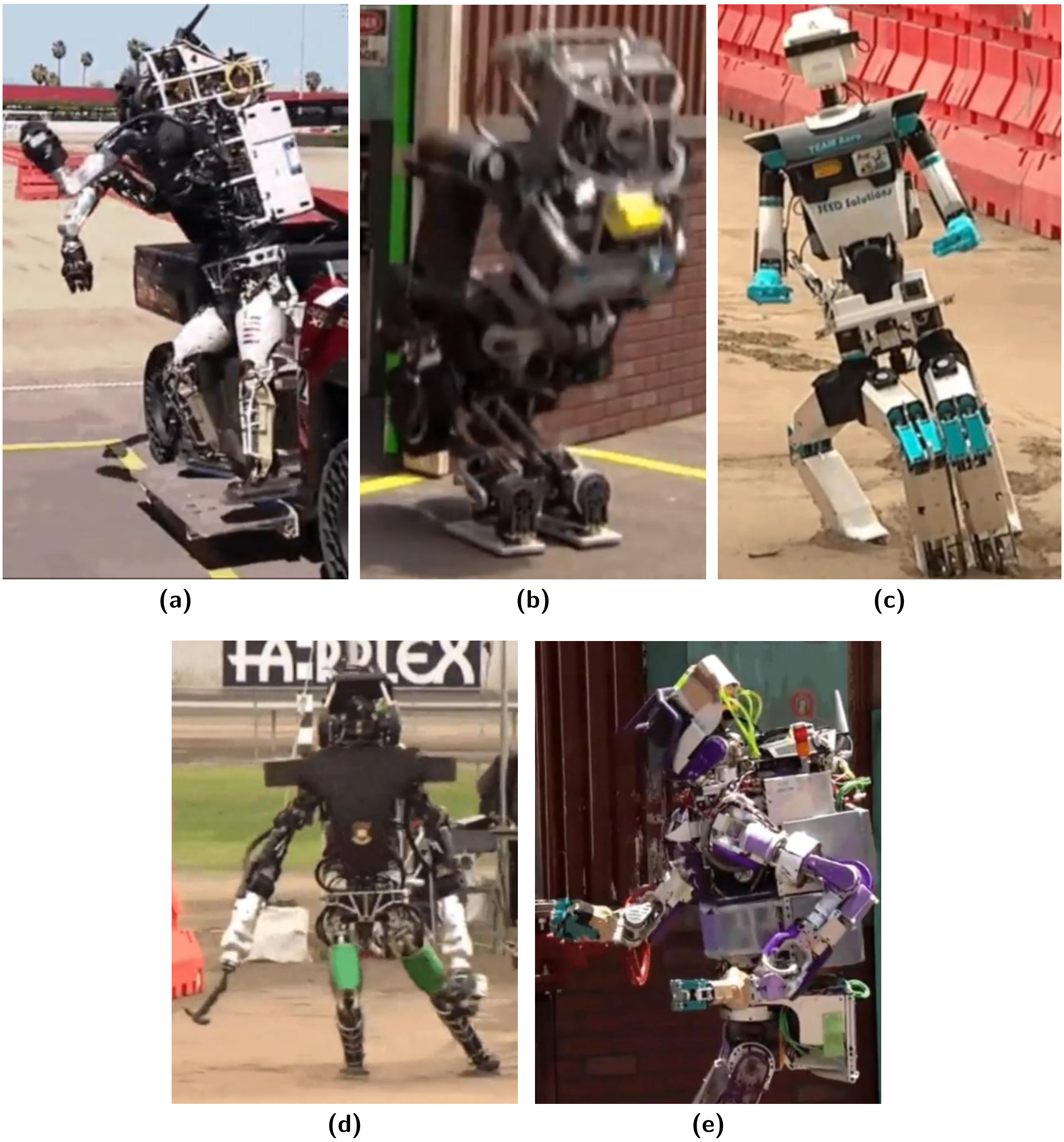


Figure 1-2: Fall overs of different humanoid robots during the 2015 DARPA robotics challenge and their sources were as follows: (a) software bug, (b) power cut-off, (c) external disturbance, (d) real environment, and (e) operator error.

Effects of Humanoid Fall Overs

In all the above-listed fall over instigating source categories, except for the power cut-off, the extent of damage can be devastating enough to instantly immobilize the robot requiring human intervention. However, the exact amount of damage inflicted on any humanoid depends on the source, falling over configuration, impact absorbing capacity of the robot's body parts, terrain nature, etc. In the case of humanoid fall overs due to power cut-off, the extent of damage is relatively less since the humanoid collapses vertically with all the joints controlled in a passive mode. From the energy point of view, during the vertical collapse the potential energy (PE) of the system is reduced and also less PE gets converted to the kinetic energy (KE) just before impact resulting in less damage.

In the fall overs initiated by any of the sources from 1 to 5, the joints of the humanoid remain under active control and its state can be controlled in an underactuated manner. This gives the possibility of applying any control measures to reduce the damages. While in the case of 6 and 7 fall over sources, not all joints are under active control and hence the effectiveness of control measures are significantly limited by the extent of hardware or communication failure observed in the system. In the case of power cut-off, there is absolutely zero control over the robot, however, as stated earlier, the damage due to vertical collapse is relatively less when compared to the other fall over sources. In this work fall overs initiated by means of external disturbances has been considered. However, the proposed techniques can still be applied for other fall over sources where the humanoid remains in active control.

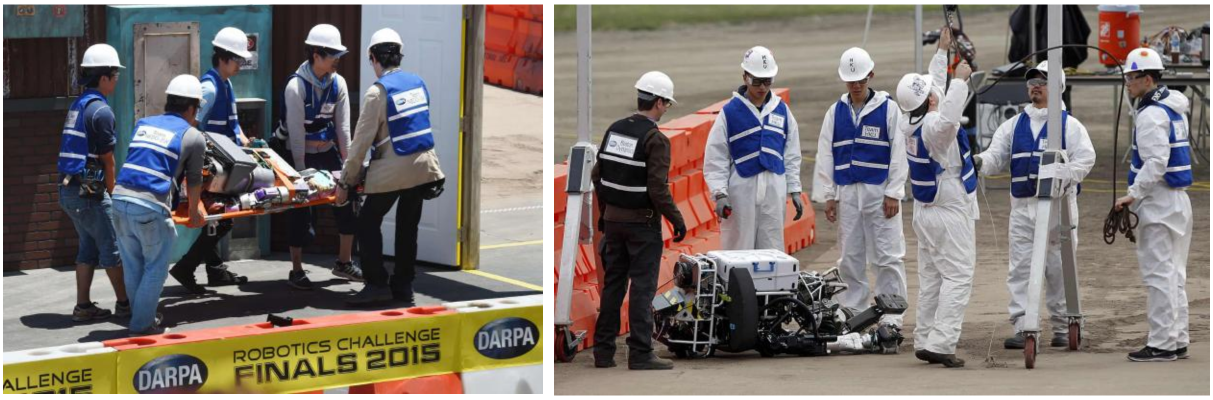


Figure 1-3: Pictures show the devastating effect of humanoid fall overs and as a result humans had to intervene to carry them away.

1-3 How to Deal With Humanoid Fall Overs?

As stated earlier, for humanoids to operate autonomously in man made environments without any human intervention it is necessary for them to deal with the fall overs. To effectively address this inevitable issue of humanoid fall overs, it is subdivided by the robotics research community into three major components: fall prediction, controlled fall, and fall recovery. So the path to recover from any fall over in to order to successfully operate in real environments is shown in Fig. 1-4. The success of each component depends on the successful completion of its previous component. For instance, if the robot has to recover autonomously after a fall over it is paramount for the robot to sustain minimum damages such that it remains in operable condition to recover. If the robot has to sustain minimum damages then the fall over should be controlled in such a way that the robot adopts an optimal configuration to mitigate the impact forces during its ground contact. Although, the controlled fall minimizes the impact forces, it is very difficult to reduce the damages sustained to zero. Hence, this calls for a method which can predict the inevitable fall over of humanoids, which in turn can initiate the controlled fall actions. To summarize, a successful recovery of a humanoid depends on how good the controlled fall component is in minimizing the impact forces and this in turn depends on how early the fall prediction component can predict the inevitable fall over of humanoids. In the present work, the first two components are analyzed in detail and appropriate techniques are proposed to address them effectively.

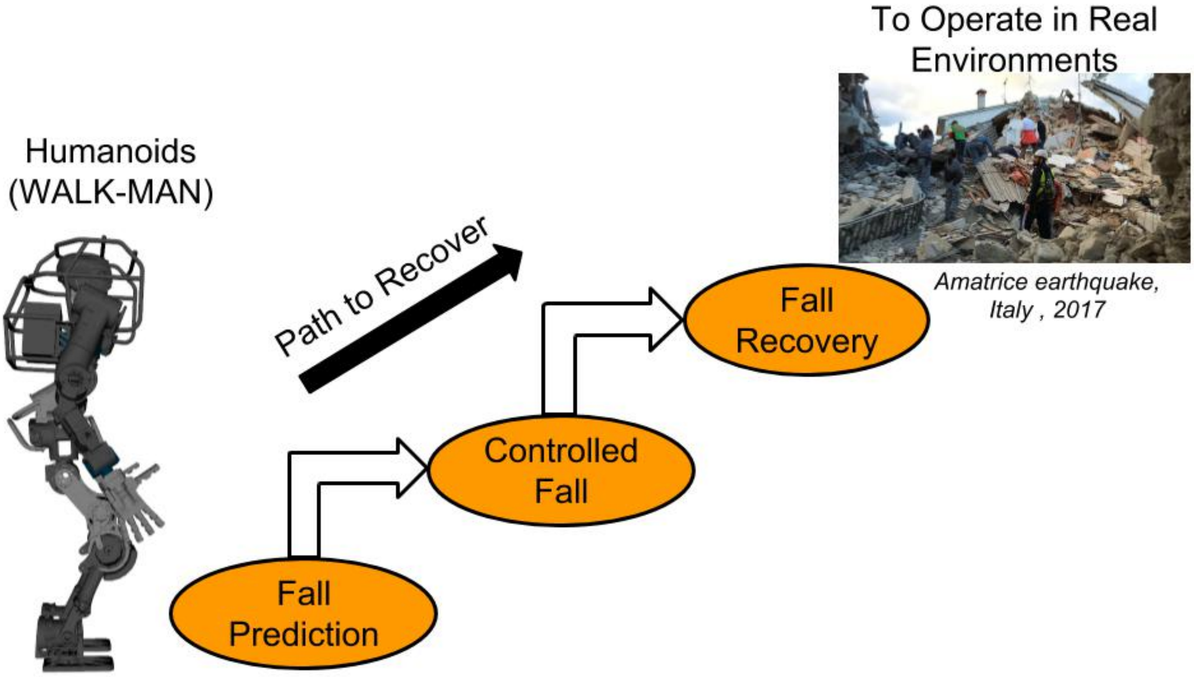


Figure 1-4: The recovery path for the humanoids to deal with the inevitable fall over issue to operate autonomously in real environments.

Chapter 2

System Overview

In this chapter, the different humanoid platforms used in this work for the development and verification of the proposed techniques are introduced here. This chapter also discusses the different sensor modules integrated with the system and the software architecture used to communicate between the system and sensor modules.

2-1 Humanoid Systems

In the presented research work two different humanoid systems are used for the evaluation and verification of the proposed techniques.

WALK-MAN

Whole-body Adaptive Locomotion and Manipulation (WALK-MAN) is a humanoid platform developed under the European Commission project with an objective to operate it in buildings that were damaged due to either natural or man-made disasters. In particular, it was designed and developed with special focus on three features: dexterous and powerful manipulation skills for intricate tasks like valve turning, opening door, etc.; robust whole-body locomotion and manipulation to maneuver safely in a cluttered environment; and physical sturdiness to operate man-made tools like pneumatic drills, cutters, fire extinguishers, etc. WALK-MAN with powerful, yet soft and compliant actuators gives it the natural adaptability, intractability, and sturdiness. The WALK-MAN humanoid system is shown in the Fig. 2-1a

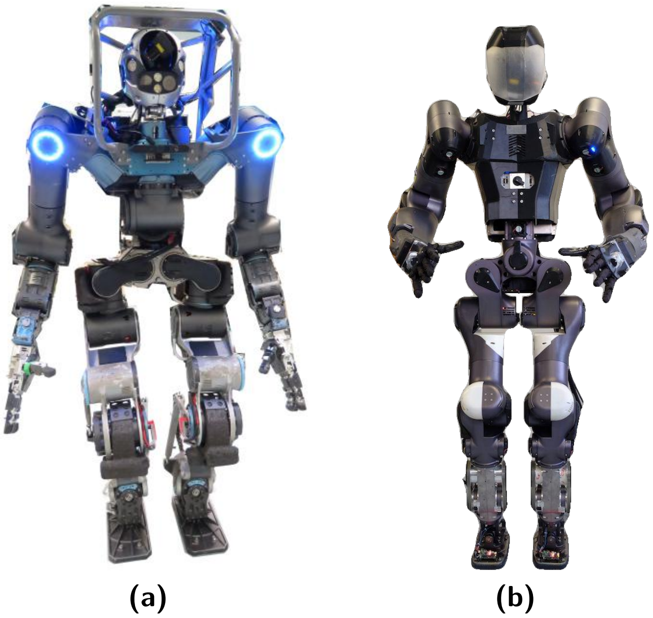


Figure 2-1: The two different humanoid systems used in this work are shown here: (a) WALK-MAN, and (b) CogIMon.

CogIMon

Cognitive Interaction in Motion (CogIMon) is a relatively lightweight humanoid robot developed under the European Union (EU) research project in the Horizon 2020 work programme. Figure 2-1b shows the CogIMon humanoid system. The main aim of the project is to modify the way humans and robot interact by regulating their motions and behaviors in an active and adaptive manner. The humanoid platform is developed to achieve the aforementioned objective by including whole-body impedance actuation, adaptability, prediction, and flexibility. While WALK-MAN has 31 degrees of freedom (DoF) and weighs 140 kg, CogIMon has 29 DoF and is much lighter than the WALK-MAN weighing 64 kg. Also, compared to the former, the latter lacks the pitch joints for hip and neck making the total number of joints in CogIMon to be 29. Another notable feature of CogIMon humanoid is its parallel kinematics ankle which permits the distribution of the power of the actuators to the two DoF of the ankle taking full advantage of the available power capacity of the two actuators.

2-2 Sensor Modules

The sensor modules which are integrated with each of the humanoid systems are presented briefly in the following subsections.

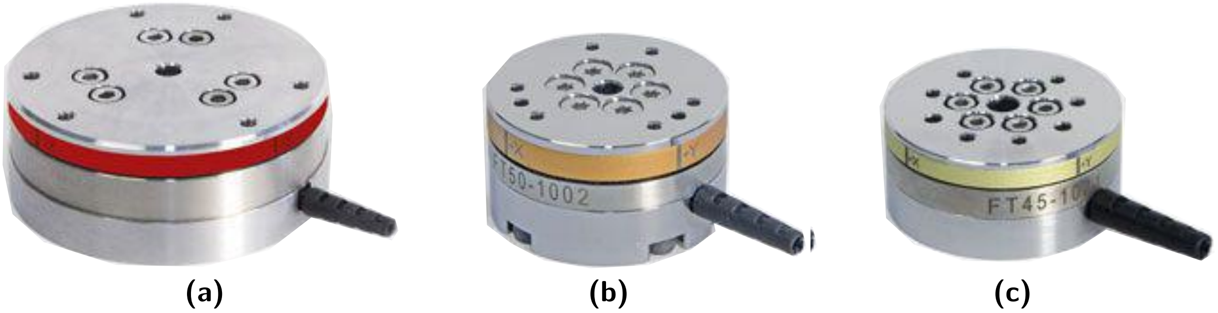


Figure 2-2: Different versions of customized force torque sensors used for the humanoid systems: (a) FT-80, (b) FT-50, and (c) FT-45.

2-2-1 Force/Torque Sensor

A series of customized 6 DoF force/torque (FT) sensors are integrated at the ankle and wrist joints of both the robots shown in the previous section. The working principle of FT sensors is based on a three-spoke structure which includes six pairs of semiconductor based strain gauges which measure the load on each cell induced due to the load applied. The sensor also includes the necessary data acquisition and conditioning electronics and it communicates with the other system components using the EtherCAT communication protocol. Three different FT sensors, FT-80, FT-50, and FT-45 shown in Fig. 2-2 are used with the humanoid systems. While FT-80 and FT-50 are used for the ankle and wrist joints of the WALK-MAN humanoid, FT-50 and FT-45 are used for the CogIMon system respectively. Some of the specifications of the sensors are tabulated in Table 2-1.

Table 2-1: Specification of the FT sensors

FT sensors	Specifications
FT-80	dimensions (D×H): 88×34.5 mm, load range: (F_{xyz}/M_{xyz}): 3.5, 2.5, 2.5 kN / 200 Nm, resolution (F_{xyz}/M_{xyz}): 900 mN / 50 mNm, data rate: 1 kHz.
FT-50	dimensions (D×H): 50×28.0 mm, load range: (F_{xyz}/M_{xyz}): 2.0, 1.0, 1.0 kN / 40 Nm, resolution (F_{xyz}/M_{xyz}): 500 mN / 10 mNm, data rate: 1 kHz.
FT-45	dimensions (D×H): 45×20.8 mm, load range: (F_{xyz}/M_{xyz}): 1.0, 0.5, 0.5 kN / 20 Nm, resolution (F_{xyz}/M_{xyz}): 250 mN / 5 mNm, data rate: 1 kHz.

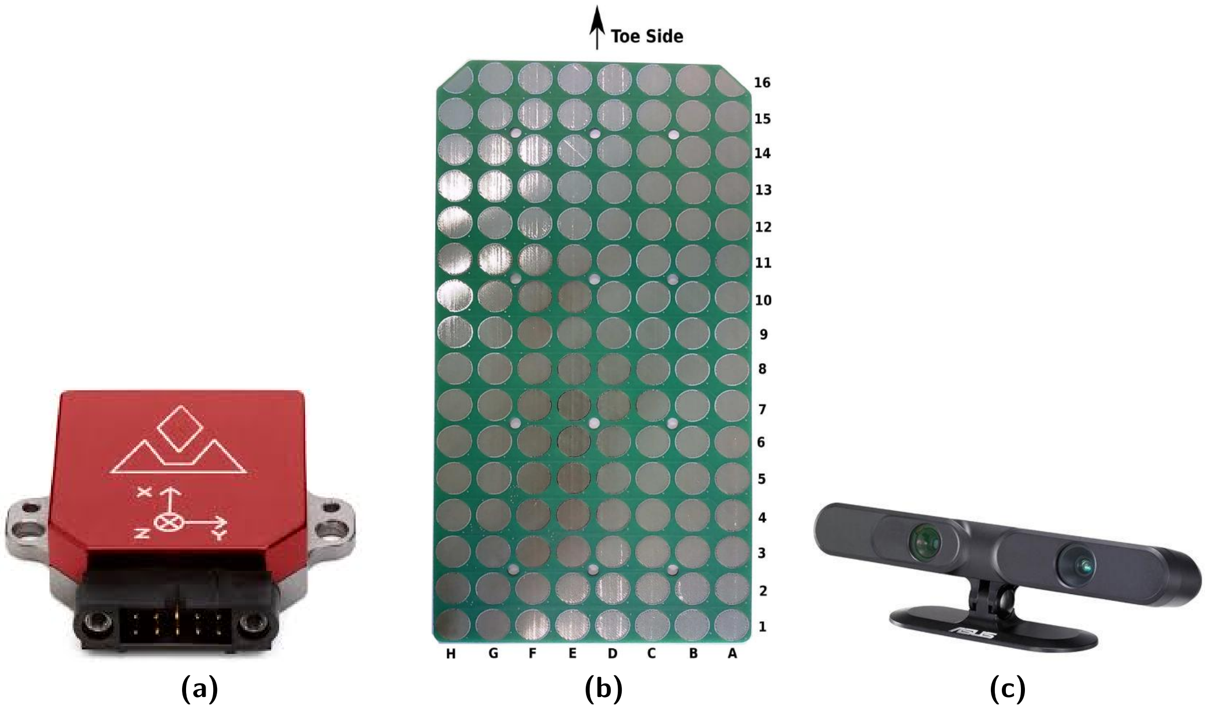


Figure 2-3: Different sensor modules which are integrated and used along with the system are shown here. (a) IMU of VectorNav, (b) foot pressure sensor of IIT, and (c) RGB-D sensor of ASUS.

2-2-2 Encoders

An encoder is an electro-mechanical device used to measure the absolute position of any actuated joints of the robot. For WALK-MAN and CogIMon humanoid systems two sets of position encoders have been used. One is attached after the harmonic gear box to measure the position of the motor after gear arrangement. The other one is located after the series elastic bar to determine the link side angle. The encoders used here are absolute position sensors with 19-bit resolution and outputs the data at 1kHz.

2-2-3 Inertial Measurement Unit (IMU)

An IMU shown in Fig. 2-3a is an electronic device attached to any physical system to measure its rate of linear acceleration using accelerometers, rate of change in rotational axes (roll, pitch and yaw) using gyroscopes, and it also includes a magnetometer to account for the changes in its orientation due to the drift. For the humanoid systems considered here three IMUs are used in total. One is attached to the base/waist link of the robot and the other two are fixed to each foot of the robot. The former is used to get the orientation, angular velocity, and linear acceleration of the waist, while the latter IMUs are used to correct the robot's orientation.

2-2-4 Foot Pressure Sensor (FPS)

External proprioceptive sensors are essential as they give more reliable information about the contact, such as its area (point or surface), pressure distribution, type of terrain, etc. We believe that this information is important in determining the current balance state of the robot. Most of the present robots lack foot sole pressure sensors, so we have developed a resistive foot sensor as shown in Fig. 2-3b, which is placed under each foot sole of the humanoid robot. The sensor is arranged as an array of 128 resistive based pressure cells, each of 16 mm in diameter, spaced across a layout of 160×320 mm. Upon application of pressure, each cell generates an analog output (0-255 V), sampled by an 8-bit data acquisition system at 800 Hz.

2-2-5 RGB-D Sensor

Vision also plays an equally important role in estimating the position and orientation of the robot with respect to the environment. In our system, an ASUS Xtion Pro RGB-D sensor shown in Fig. 2-3c, mounted on the robot's torso, is used for generating depth images at a rate of 30 fps. From the streaming images, 50 features are randomly extracted and tracked in successive frames. For feature extraction, the Shi and Tomasi's [1] algorithm is used, and for tracking, the iterative Lucas-Kanade method [2] is utilized. Under a disturbed condition, e.g., the robot's falling over, the environment that is visualized through the RGB-D, undergoes rapid changes, and this can be quantified and actively monitored by constructing the 3-dimensional (3D) optical flow of the features. During the normal operation of the robot, the change in flow velocity is very gradual, unlike the fall cases. This is exploited in this work for predicting the fall. To get the 3D optical flow of the features, their respective locations in the real world are obtained from the 3D point clouds, and their velocities are computed by differentiating their position in successive frames over the frame rate time.

Table 2-2 shows the specifications of the different sensor modules used along with the system.

2-3 Software Components

2-3-1 Simulation Control Architecture: Gazebo + ROS

The simulation control architecture adopted to numerically simulate the robot models to evaluate the different controllers proposed in this work is shown in Fig. 2-4. The architec-

Table 2-2: Specification of the Sensor Modules

Sensor Modules	Specifications
RGB-D Sensor	VGA(640×480): 30 fps, range: 0.8-3.5 m, resolution: 640×480 (ASUS).
Foot Pressure Sensor	Size: 0.16×0.32 m, 128 pressure cells, analog output rate: 800 Hz (developed in IIT)
IMU	Range: Heading, roll +/-180°, pitch: +/- 90°, output rate: 800 Hz (VectorNav VN100)
Joint encoders	Absolute position sensor with 19-bit resolution, output rate: 1 kHz (IC-Haus/Ballu)

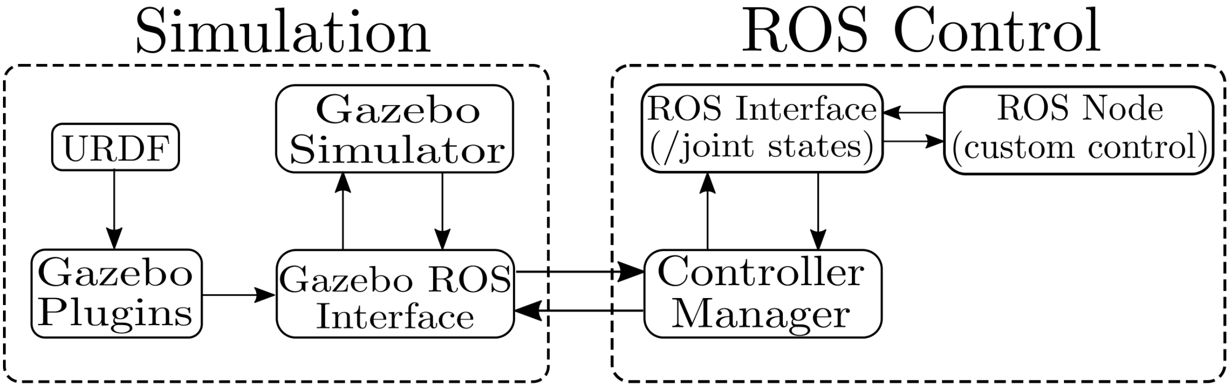


Figure 2-4: Simulation control architecture used to simulate the robot models and it involves the simulator Gazebo and the middleware ROS.

ture involves six major modules and the function of each one is explained briefly below:

- Gazebo Plugins: This module takes the robot model given in the form of universal robot description format (URDF) as input and loads gazebo and ros related plugins such as the low level joint control plugins, different sensor related plugins, etc.

- Gazebo ROS Interface: This serves as the common interface to send and receive data between gazebo and ROS. The appropriate interfaces for joint position, velocity, and effort or to the sensors are created with the plugins loaded in the previous module.

- Gazebo Simulator: It simulates the loaded robot models with the commands received from the Gazebo ROS Interface module, computes the next state of the model with a robust physics engine and sends back the next state of the robot. With high-quality graphics it ensures proper visualization of the simulated model and with a programmable user interface it is easier to interact with it dynamically.

- Controller Manager: The objective of this module is to load, unload, and update all the controllers depending on each controller's publishing rate. It sends the low level control commands to the simulator through the gazebo-ros interface and reads the state of the robot through the same. The low level control commands are computed with the high level commands generated by custom controllers in the ROS Node module.

- ROS Node: Custom controllers developed by the user to generate high level commands for the robot to carry out particular task are run as nodes. These are nothing but executables which ROS uses to communicate with other similar nodes.

- ROS Interface: It is a communication medium which ROS uses to communicate with nodes and control managers. The communication is either in the form of messages which are one-way or through the use of services which are two-way mediums.

2-3-2 Real-time Control Architecture: XBotCore

The WALK-MAN humanoid robot used for the evaluation of the proposed fall prediction method is powered by the *XBotCore* (*Cross-Bot-Core*), a Real-Time (RT) software platform for robotics. *XBotCore* is an open-source platform[1], designed to be both an RT robot control framework and a software middleware. It satisfies hard RT requirements while ensuring a 1 kHz control loop even in complex multi-DoF systems. The *XBotCore* application programming interface (API) enables an easy transfer of developed software components to multiple robot platforms, inside any robotic framework or with any kinematics/dynamics library as a back-end. Out-of-the-box implementations are available for the *YARP* and *ROS* software frameworks, and for the *RBDL* and *iDynTree* dynamics libraries.

A robot hardware abstraction layer (R-HAL), permits to seamlessly program and control any robotic platform powered by *XBotCore* [3]. Moreover, a simple and easy-to-use middleware API for both RT and non-RT control frameworks is available. The *XBotCore* API is completely flexible with respect to the external control middleware the user wants to utilize. The framework gives access to a wide set of motors and sensors both in simulation (using the Gazebo[2] environment) and on the real robot, which let the users to easily port their code from the simulation to the hardware without any code modifications.

As shown in Fig. 2-5, *XBotCore* spawns three threads in the Linux Xenomai[3] Real-Time Operating System (RTOS):

- The R-HAL RT thread which runs at 1 kHz is responsible to manage and synchronize the EtherCAT slaves on the robot, i.e., the electronic boards of the motor control and sensors data acquisition.

[1] https://github.com/ADVRHumanoids/XBotCore

[2] http://gazebosim.org/

[3] https://xenomai.org

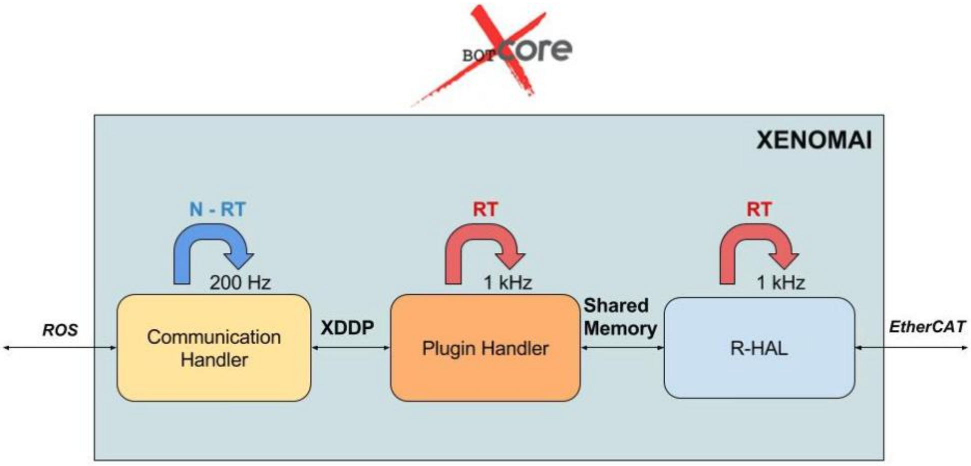


Figure 2-5: *XBotCore* threads and communication architecture.

- The Plugin Handler RT thread which also runs at 1 kHz is responsible to start a set of loaded plugins, execute them sequentially, and close them before they are unloaded. It is possible to dynamically load and unload one or more plugins using the plugin handler. A shared memory communication mechanism is used to share data between this component and the R-HAL at 1 kHz.

- The Communication Handler, a non-RT thread runs at 200 Hz, and it is responsible for the communication with the external frameworks. This component provides the option to send the desired robot state from the non-RT API to the chosen communication framework and receive the references respectively. The Communication Handler uses cross domain datagram protocol (XDDP) for the asynchronous communication between RT and non-RT threads, guaranteeing a lock-free inter process communication (IPC). The run loop of this component is quite simple: it updates the internal robot state using the XDDP pipe with the non-RT robot API, sends the robot state to all the communication frameworks, receives the new reference from the requested 'master' (we avoid to have multiple external frameworks commanding the robot) and finally, sends the received reference to the robot using the XDDP non-RT robot API.

2-3-3 Robot Models

Two different model types of the system are considered here: First is the complete 3D numerical models of the robot which are used to evaluate/verify the performance of the proposed controllers, and Second is a set of simplified models which are generic enough to represent any humanoid system. The latter models are developed to portray certain situations succinctly and used to plan online and generate control actions for humanoids in a time efficient manner.

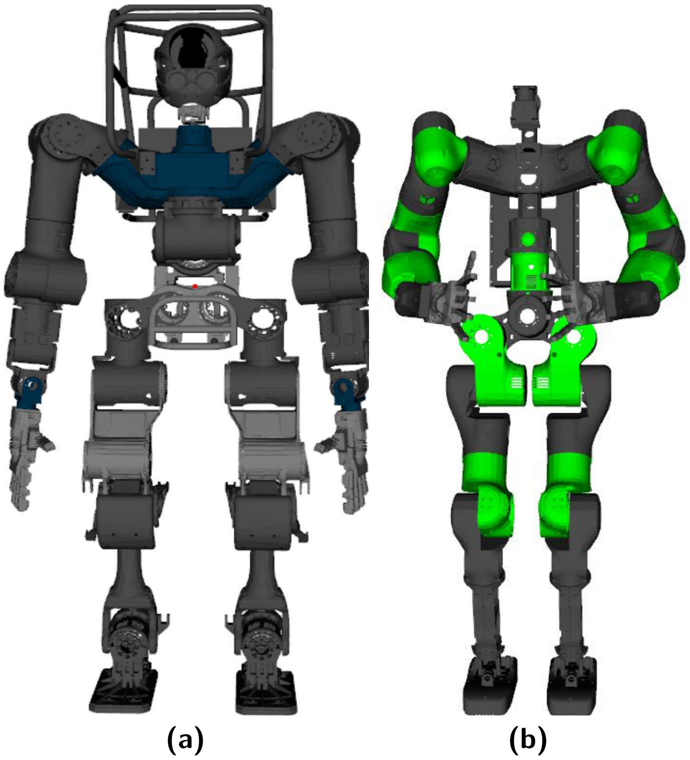


Figure 2-6: The numerical models of the humanoid systems used in this work to evaluate different controllers are shown here: (a) WALK-MAN, and (b) CogIMon.

3D Models

To generate whole body control motions for the robot and to analyze the performance of the proposed controllers an approximate 3D numerical model of the humanoid systems are developed and used throughout this work. The models are generated in the form of URDF considering the kinematic constraint such as link length, type of joint, its position and velocity limits. The model also includes some dynamic properties such as the inertia of each link, joint torque limit, damping and friction of the joints, etc. However, since the last two parameters are difficult to determine even with the real system they are included here to only represent the system's behavior approximately. The numerical models of the humanoid systems which were generated as stated earlier are shown in Fig. 2-6. The 3D models of WALK-MAN and CogIMon humanoid systems are shown in Fig. 2-6a and Fig. 2-6b respectively.

Simplified Models

The following are the two simplified models used primarily in the controlled fall part to make some efficient control decisions.

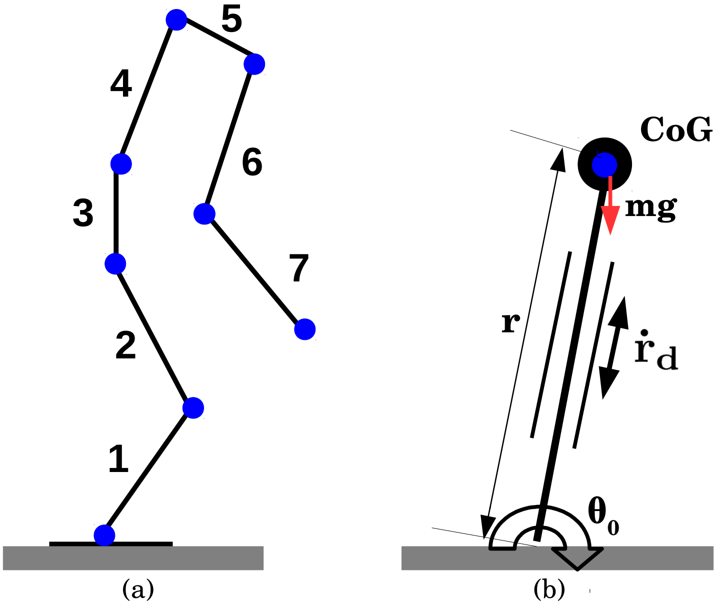


Figure 2-7: Simplified models of the humanoid systems: (a) segmented planar robot and (b) telescopic inverted pendulum model.

- **Telescopic Inverted Pendulum**: The telescopic inverted pendulum (TIP) model shown in Fig. 2-7b is the simplified model used to represent the fall over of humanoids in general. The point mass m denotes the robot's mass, CoG is its center of gravity, and the mass less rod of length r represents the distance between center of gravity (CoG) and the tipping point of the humanoid robot. The telescopic/sliding action along the rod enables it to contract or expand and this is used to model the squatting and extension motion of the humanoid. This is used in Section 11-2 to generate optimal control input u in the form of desired sliding velocity $\dot{r}_d$ during the fall over of humanoids.

- **Segmented Planar Robot**: The segmented planar robot (SPR) model shown in Fig. 2-7a is a relatively less complex planar model of the humanoid used to represent the rolling posture of humanoids in the form of polygons in a 2-dimensional (2D) plane. The model shown in the figure has been constructed with the contact points in interest and it has 7 links with each link representing a particular part of the humanoid. Links 1-7 denote the foot and lower-, upper-leg, waist, torso, torso-shoulder connector, upper-, and lower-arm, respectively. The mass and length of each link are taken equivalent to the respective part of the humanoid system considered. This model is used in Chapter 12 to generate an optimal rolling configuration for humanoids during their fall over.

Part II

Fall Prediction

Chapter 3

Motivation, Related Works, and Challenges

3-1 Why Focus on Fall Prediction?

As stated in Section 1-3, to address the issue of humanoid fall overs it has been subdivided into three major components: fall prediction, controlled fall, and fall recovery. The success of each component depends on the success of its previous component. Fall prediction being the primary component, predicts the inevitable fall over of humanoids, which is followed by the controlled fall motion to mitigate the impact forces during ground contact, and the fall recovery actions to recover from it. For both the controlled fall and the recovery actions to be successful it is prominent to predict the imminent fall over of humanoids and this has to be done as early as possible to increase the success rate of its following components. The effect late fall prediction will have on the performance of controlled fall is demonstrated in Appendix A-4. Also, it is necessary for the fall prediction component to perform reliably on different terrains, disturbances, robot configurations, etc. All the aforementioned things calls for more focus on fall prediction.

3-2 Related Works

In this section, we briefly summarize the literature related to falls in general on humans and humanoid robots. We first list out the existing traditional balance measures for robots in Section 3-2-1 and discuss why they are not sufficient in predicting the imminent fall over of humanoids. Next in Section 3-2-2, we introduce some of the prominent measures or techniques proposed mainly to predict the fall over of humanoids and shortly highlight their working principle and drawbacks in handling generic scenarios. Since our method is inspired by humans, a quick overview of the fall works related to humans is made in Section 3-2-3, mainly focusing on the similarities and differences when compared to the humanoid falls. Finally, in Section 3-2-4, we succinctly explain why and how the Kalman filter is used in this paper.

3-2-1 Balance Measures for Humanoids

A biped robot being a floating base and an underactuated system, maintains its postural balance by means of unilateral contacts with the ground through its feet. It maintains constant foot-ground contact during a static case such as standing, and it makes sporadic foot-ground contacts during dynamic motions such as walking. In both the cases, the overall stability of the system has been attained by controlling its postural stability through the robot's foot-ground contact. To monitor and control the stability of the humanoid system, several balance measures have been proposed over years of research. The prominent balance measures used are as follows:

- Ground projection of the center of mass (GCoM) [4]: a static balance measure which states that the robot is stable if the gravity line of action of the robot's center of mass (CoM) fall within the convex hull of the feet. The exit of GCoM from the convex hull denotes an uncompensated moment acting on the robot to tip over (likely) its convex hull edge.

- Center of pressure (CoP)[5]: it is a point on the foot-ground surface where the resultant ground reaction force actually acts. If the CoP remains within the support polygon of the feet the robot is stable, and its movement to the edge denotes the robot's instability.

- Zero moment point (ZMP): a point on the ground where the moment generated by the gravity and inertial forces equals zero. On the assumption of an unilateral foot-ground contact, the zero moment point (ZMP) is similar to CoP, and hence they follow the same stability criterion. Whereas, if the contact is taken to be glued, then the ZMP can leave the support polygon, unlike the CoP. This can be used to propose an instability criterion [6, 7].

- Foot rotation indicator (FRI) point [5]: it is a point on the foot-ground contact surface, within or outside the convex hull of the foot support area, at which the resultant moment of the force/torque impressed on the foot is normal to the surface. While FRI point inside the support polygon denotes the robot's stability, the point being outside represents the one's instability.

- Zero rate of change of angular momentum (ZRAM) point [8]: the point on the ground where the resultant ground reaction force should act such that its line of action passes through the robot's center of gravity (CoG), resulting in zero centroidal angular momentum. The instability of the robot is proportional to the distance between the ZRAM and CoP points.

- Capture point (CP) [9]: it is a point on the ground where the robot can step in order to bring itself to a complete stop. If the CP is within the convex hull of the feet the robot is considered to be stable, if it is outside the hull but within the reachable workspace of the robot's feet then it is considered to be stabilizable, and in situations where it is outside both the hull and workspace the robot is said to be unstable and likely to fall.

The aforementioned balance measures could be a necessary condition to detect the instability of the legged robots, however, they are not sufficient enough to predict the imminent fall over of a humanoid robot in generic situations. For example, during the human's walking, especially in their toe-off and swing phases, the stability criteria of CoP/ZMP and FRI are often violated. Furthermore, each measure has it own drawback; for example, GCoM could only be applied for static cases, CoP and ZMP cannot quantify the degree of instability, FRI can only be applied during the single stance phase of walking, and ZRAM considers a strong assumption of flat foot surface. These limitations led the researchers to look for different measures/techniques to predict the imminent fall over of humanoids.

3-2-2 Fall Prediction in Humanoids

The ideal way to predict the imminent fall over of humanoids is to determine the set of all states which would not lead the robot to a fall over state. Such states together form a viability kernel, and this can be used to compute an intuitive and reliable distance between fall and non-fall states. However, in reality, it is computationally expensive to obtain such kernel, considering the humanoid's non-linearity, under-actuation, and its generic operating conditions [10]. The feasible option is to devise a faster measure to predict a humanoid's fall over, given its state. In this light, the following prediction methods have been proposed.

Pratt and Tedrake cite the traditional stability margins such as ZMP, CoP, FRI, and ZRAM as less general, and not suitable for predicting instabilities under all situations [11]. They propose a velocity-based technique, called the capture point (CP), to detect instabilities in humanoids [11, 12]. If CP is within the convex hull of the feet the robot is considered to be stable, if not the robot has to make a step such that its new convex hull encloses the CP. Though this measure has been predominantly successful on flat terrains, any inconsistencies in the robot's model or noises in the sensors affect the performance of CP considerably. To overcome this, a Kalman filter based CoM estimator with an offset is proposed, which is used to compute a reliable CP called corrected capture point (CCP) [13]. The offset accounts for any discrepancies of the model or sensor noises or a combination of both. Another shortcoming of CP is its inability to handle multi-contact poses, which has been overcome by proposing zero step capturability based on the robot's centroidal dynamics [14]. The authors successfully verify the proposed method's improved performance when compared to conventional CP in multi-contact scenarios. In the above methods, the distance between CP and the closest edge of the support polygon is used to quantify the degree of stability/instability. In contrast, a scalar integer control variable based on the total energy of the mechanical system was proposed in [15], but its success was considerably limited to specific configurations of the robot.

All the aforementioned methods use different simplified models such as inverted pendulum, linear inverted pendulum or a point mass model. Though the use of simplified models increases the computation speed, it immensely limits these methods generic applicability to different configurations of the robot, varied terrains, etc. On the other hand, in this paper, the proposed method considers the complete robot model and the information extracted from various sensors inspired by humans. Hence, it is not restricted to any specific configuration or terrain, which makes our method more versatile to handle generic situations.

There are also others who proposed much simpler and computationally least expensive methods by setting static thresholds on the tilt angle of the robot and its derivative or the CoM position of the robot for predicting the fall successfully [16, 17, 18]. Though these techniques have been quite successful in certain cases, they have inherent drawbacks such as *late fall prediction* (LFPr) and *false positives* (FP), which are reported in Chapter 6. This is mainly due to the assumption of a hard boundary, over-reliance on a single sensor data/feature and negligence of sensor noises.

Apart from the above works, there are few research works reported for predicting falls in dynamic cases (walking in particular), which are based on the deviations from an online/offline constructed sensor model [19, 20]. Some works adopt the technique of devising a feature vector, which includes stability margins, velocities, and contact forces, and later monitoring its deviations for fall prediction [21, 22]. This was further extended to build predictor models with probabilistic [23] and machine learning techniques [24]. Since our work presented here is also feature based with each feature extracted from multiple sensors, the threshold extraction process and the performance measures used for evaluating the prediction technique are inspired partly by the above works.

The notable difference compared to the above works is the inclusion of newly proposed FPS and vision sensor, and together with the conventional sensors (joint encoders and inertial measurement unit (IMU)) provide much detailed information about the state of humanoids akin to the balance sensory system of humans. This helps in making a reliable and agile prediction of humanoids fall over. Also, the sensor redundancy, along with our online interpolation and impulse adaptation techniques, reduces the number of training sets and lets the prediction method more generic in terms of disturbance magnitude and its direction. Further, unlike using static thresholds directly, we propose cost functions to assign cost depending upon each variable's distance from their respective thresholds. This makes the prediction more reliable, less erratic, and limits the occurrences of LFPr and FP. Finally, with Kalman filter based multiple feature technique, the method's ability to make optimal decisions in uncertain situations such as different terrains, noise levels, etc., is also incorporated.

3-2-3 Fall Detection in Humans

Falls have been a major concern also for humans, as they can result in bone fracture, loss of independence, or even death in worst scenarios. It is more common with old people since their body undergoes numerous physical changes due to the age, making them weaker and more susceptible to falls. Unlike humanoids, in the case of humans, the fall research is more dedicated towards detection rather than prediction. Since early fall detection helps in providing timely medical assistance, which can even save lives, fall over is being actively researched for humans as well. The fall detection system for humans can be broadly categorized into external and wearable sensors [25]. In the former case, camera, pressure, and infrared sensors are used, while in the latter, camera and IMU are generally used. A notable work is the automatic fall detection and activity classification using a wearable smart camera proposed by Ozcan [26]. Though the idea of using cameras for fall is similar to our system, in the reported work it is used for detection by means of correlating the histogram of edge orientations and their strength values. In our work, we are using the vision sensor for predicting falls by monitoring the optical flow of selected features in each image frame.

3-2-4 Kalman Filter Based Sensor Fusion

Predicting fall reliably under the influence of different level of noise in each sensor is extremely difficult. To manage this challenging task, a Kalman filter based multiple feature technique has been adopted here. Kalman filter, introduced by Rudolph E. Kalman in 1960 [27] as a recursive solution to the discrete-data linear filtering problem, gives an accurate estimate of unknown variables using a series of measurements taken over time and its process model containing statistical noise. This filtering technique has been adopted and used widely in varied fields, such as aerospace, computer vision, medicine, economics, etc. In robotics community, the main applications of Kalman filter have been motion planning and control [28], trajectory/parameter optimization [29], and state estimation [30, 31]. In our present work, we have used Kalman filter for two major reasons: to fuse the features obtained from different sensors to a single positive integer variable, called the fall indicator, which is used to predict the probability of humanoid's fall; and lastly to estimate the above fall indicator value optimally in the presence of various uncertainties, such as different level of noise in each sensor, varied terrains, etc.

3-3 Requirements and Challenges

In general, the main requirements of a fall over prediction for the humanoid are as follows:

- generality to cope with disturbances applied in arbitrary directions and varied terrains;
- robustness to different level of noises observed in various sensors;
- agility to facilitate swift control actions;
- reliability to minimize the failure rate; and
- versatility to handle different dynamic movements.

A notable challenge in the above requirements is to balance its reliability and agility. While increasing its agility provides enough fall over lead time, it can increase the false alarm rates (failure). Conversely, decreasing it makes the prediction algorithm more reliable, however, it reduces the lead time considerably to affect the execution of controlled fall actions, which in turn can sabotage the recovery of humanoids. Hence, it is necessary to balance both reliability and agility of the algorithm to achieve better performances. Furthermore, to the authors' best knowledge, there has been no literature evaluating the performance of the existing prediction methods in generic situations, as we have tried to explore in this work. The aforementioned reasons make it necessary to research further on the fall prediction of humanoids.

3-4 Contributions

In this work, we strive to make contributions in threefold. First, a novel fall prediction method is proposed for the humanoid, which is inspired by the human balance sensory systems involving the use of multiple sensors. In particular, we incorporate usages of foot pressure sensor (FPS) and 3-dimensional (3D) optical flow generated by vision sensors into the fall prediction method. While the former can give reliable information about the contact, such as it's location, force, and nature, which plays a major role in defining the stability of bipeds, the latter is fairly robust to noise and agile in sensing swift destabilizing movements. These sensors are used along with the conventional sensors such as joint encoders and IMU to make reliable and agile fall predictions under different circumstances, e.g., rough terrain.

Second, the development of a generalized prediction method to handle arbitrary disturbances applied at random height and direction, with less training data sets. This is achieved by means of an online interpolation technique and an impulse adaptive counter limit. The former makes use of the training data collected for fall in specific directions and generalizes them to handle other intermediate direction falls. The latter uses the velocity of robot's CoG to modulate the agility of prediction, which in turn makes the method generic with respect to different disturbance magnitudes.

Last, we offer an extensive numerical analysis based on a number of features extracted from sensors, then compare the prediction performance with three different feature combinations as follows: 1) individual features, 2) a fusion of features based on performance driven weights, and 3) a Kalman filter based fusion of features. These analyses have been carried out under various circumstances including noise conditions and different terrains such as flat, uneven, tilting, and translating terrains. It then gives an insight into the pros and cons of each sensor feature, which helps in devising a suitable way to combine the multiple features to achieve the fall prediction requirements aforementioned in Section 3-3. To quantitatively evaluate and compare the performance of each sensor feature in the analyses, a set of metrics and a scalar score are proposed.

Chapter 4

Fall Prediction Method

In this section, a detailed description of the proposed fall prediction method is made. Section 4-1 addresses the main principle behind the proposed prediction method, which is followed by an introduction of the sensor features used for prediction and an overview of the complete method in Section 4-2. The difficulty in distinguishing a fall from a non-fall state, necessary parameters to counter that and their effects are briefed in Section 4-3. In Section 4-4, the procedure to determine the above parameters through fall and non-fall experiments is illustrated, and their ability to determine fall states are shown. Section 4-5 discusses an interpolation technique used to generalize the threshold extraction for intermediate directions. Finally, the fall indicator is introduced in Section 4-6, and Section 4-7 talks about how fall is predicted by monitoring its evolution.

4-1 Human Inspired Prediction Principle

The basic principle of the proposed fall prediction is inspired by how humans use their sensory systems to percept their interaction with the environment, and thereby sense their balance. The human balance control has been observed to depend on three major sensory systems (vestibular system, proprioception, and vision), which is a complex mechanism of sensorimotor integration [32, 33, 34]. The studies show that not all human sensory systems are reliable in all situations, and their noise level varies depending upon the task nature, environmental conditions, inherent defects, etc., and these uncertainties are resolved by the central nervous system (CNS) of humans by combining the sensory inputs with different weights and making them act optimally. The above three systems complement each other well in different situations, in order to provide an approximate estimation of the state. For instance, in the case of a person viewing a moving train from a stationary train, the vision errs but the other two systems correct it.

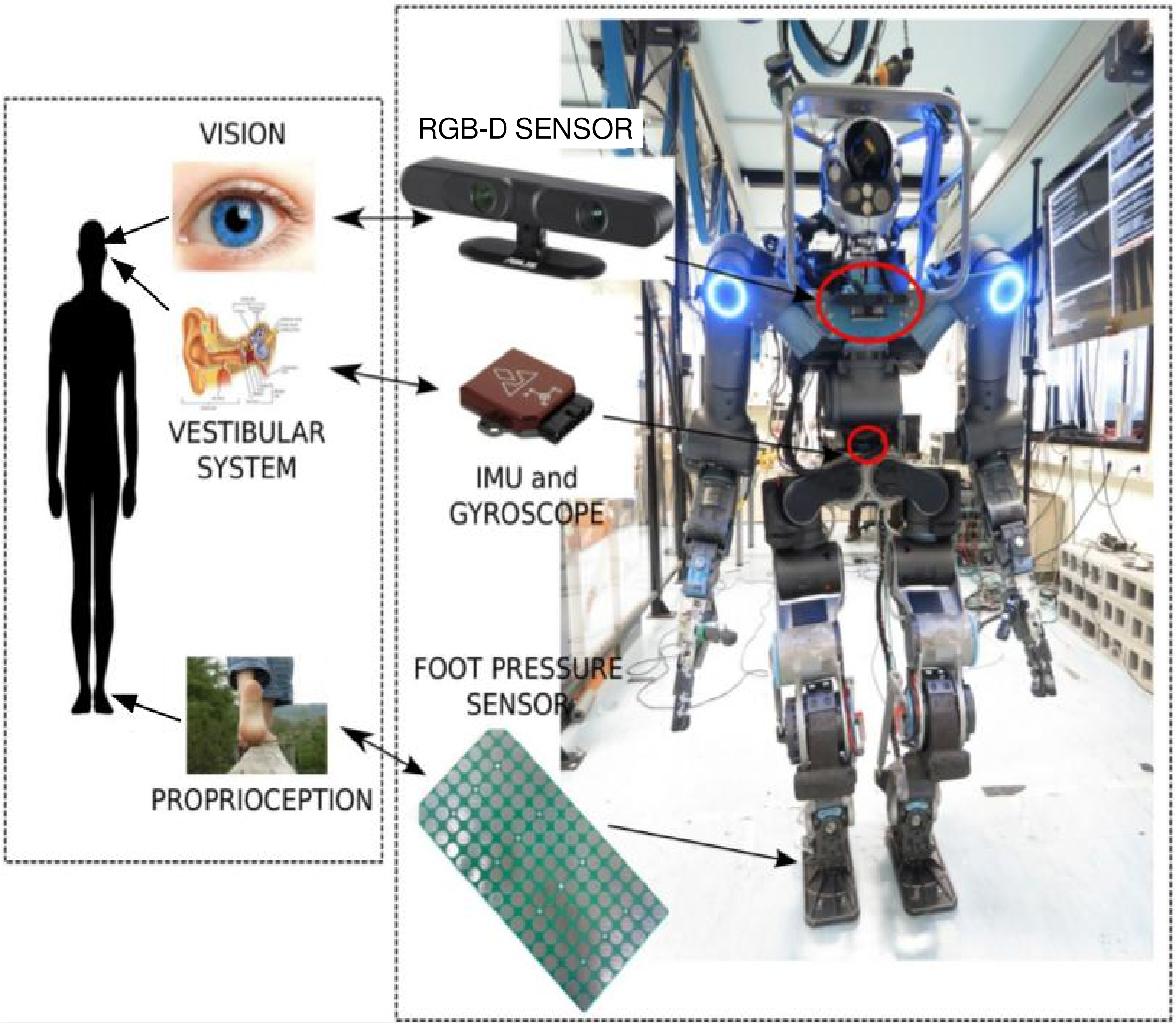


Figure 4-1: Human balance sensory systems and its equivalent sensory modalities in the humanoid system.

In this paper, we have considered the use of vestibular, proprioception and vision systems, and their respective utilities are given as follows:

- the *vestibular* system gives information about the body's equilibrium, rotational and translational movement.
- *proprioception* senses slippage at feet, contact's location, and its nature of interaction with the surface.
- *vision* perceives the relative position and orientation of the body with respect to the environment.

Table 4-1: Possible feature variables from multiple sensors

Features	Variables	N	Required Sensors
CoG	Position, velocity, and acceleration	9	IMU, joint encoders
IMU	Orientation, angular velocity, and linear acceleration	9	IMU, joint encoders
ZMP	Position	2	IMU, joint encoders
FC	Force, force gradient, contact area and its standard deviation	6	FPS
OPF	Optical flow velocity and its standard deviation	2	RGB-D sensor

Note: N denotes the number of variables.

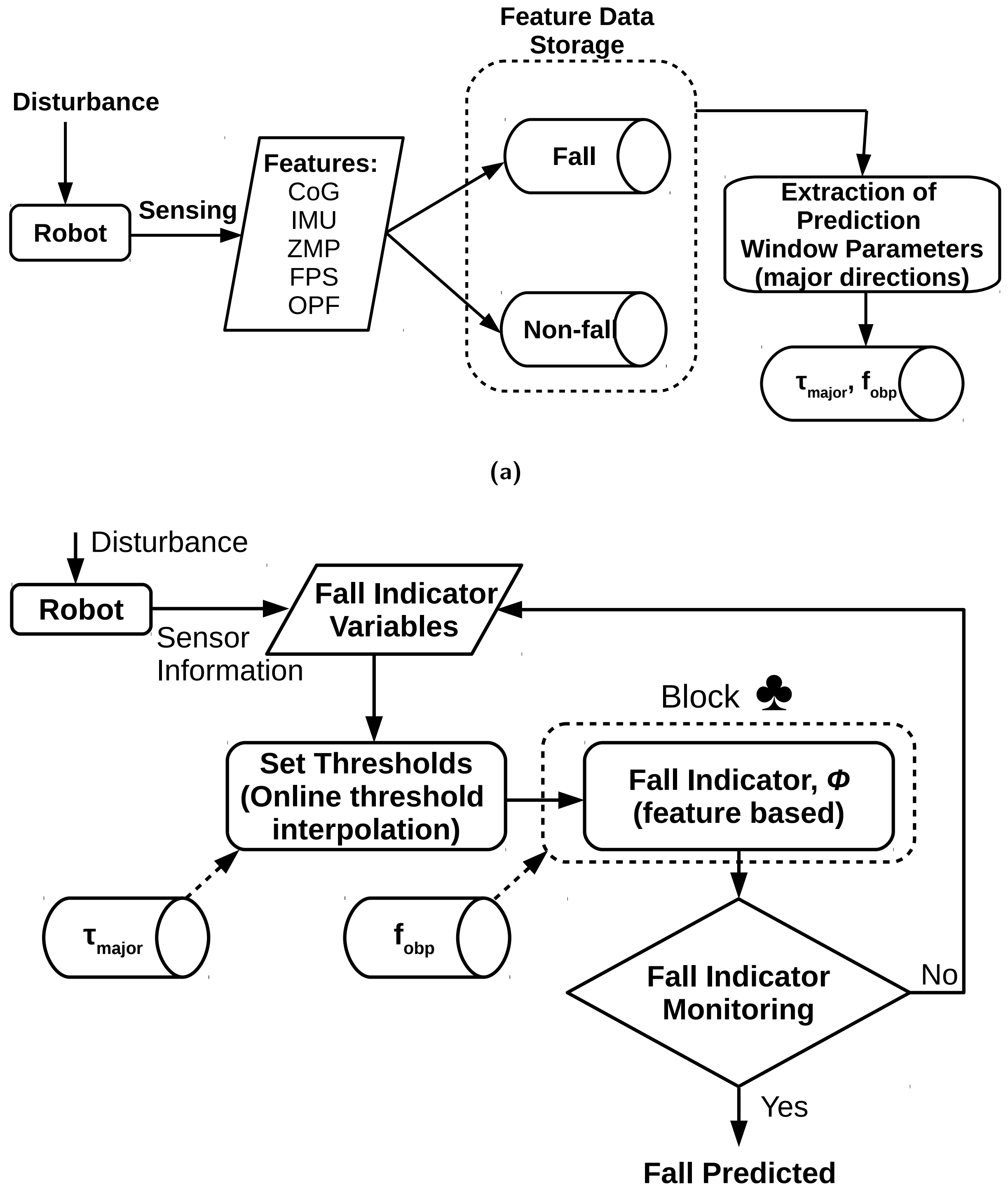
Humanoids are generally equipped with the vestibular system in the form of inertial measurement unit (IMU), which gives information about the body's orientation, angular velocity, and linear acceleration. Also, vision is available in the form of stereo cameras or light detection and ranging (LiDAR) sensors. Including foot pressure sensor (FPS) can give some additional information about its contact state, which could make the robot's absolute state estimation more reliable. Similar to the humans, the information obtained from the aforementioned sensors can be combined optimally, either by setting weights intuitively or using Kalman filter. In this paper, both the methods of fusing information obtained from the above sensors are evaluated in-depth under different conditions. Figure 4-1 shows the robot's sensory systems, which are equivalent to that of humans.

4-2 Overview of the Proposed Method

The sequential process flow involved in the proposed fall prediction method is shown in Fig. 4-2, and it is explained in detail in the subsections to follow.

The proposed prediction method is based on five different features computed from multiple sensors— center of gravity (CoG), IMU, zero moment point (ZMP), foot contact (FC), and optical flow (OPF). The possible variables from each feature can be listed in Table 4-1. Note that for the proposed falling prediction, limited sets of variables are used, which are determined based on their effectiveness to distinguish fall and non-fall cases in Section 4-4-4; these variables are called 'fall indicator variables (FIV)' in this work.

Accordingly, the whole process can be subdivided into two major parts: 1) determination of prediction window parameters for the selected feature variables, which define the segment of the data to be focused and 2) fall prediction process with the determined parameters, shown in Figs. 4-2a and 4-2b, respectively. For the former, fall and non-fall experiments are carried out to record the evolution of each feature, and finally, from the non-fall and fall


Feature Data Storage
Disturbance
Robot
Sensing
Features:
CoG
IMU
ZMP
FPS
OPF
Fall
Non-fall
Extraction of Prediction Window Parameters (major directions)
τ_{major}, f_{obp}
Disturbance
Robot
Sensor Information
Fall Indicator Variables
Block ♣
Set Thresholds (Online threshold interpolation)
Fall Indicator, Φ (feature based)
τ_{major}
f_{obp}
Fall Indicator Monitoring
No
Yes
Fall Predicted

(a)

(b)

Figure 4-2: (a) Prediction window parameters extraction process for four major directions, and (b) fall prediction with the extracted parameters, τ_{major} and f_{obp}.

data the prediction window parameters are determined for four major fall directions. For the latter, given a disturbance to the robot, the features are extracted using the sensors, and the prediction window parameters are recomputed online for the respective predicted fall direction. Then, we define a fall indicator ϕ, which is computed by observing the current values of FIV. The above computed ϕ is sent to the impulse adaptive counter limit routine to determine if the fall is inevitable or not. A positive output (fall case) from the previous module ends the process, while a negative one (non-fall case) loops it back to the features extraction step and continues. The significance of prediction window parameters is explained in the following subsections.

4-3 How to Distinguish Between Fall and Non-fall Using FIVs?

The FIV defined above are nothing but variables extracted from different features, which are used to observe the overall state of a humanoid. Though these variables possess adequate information to characterize fall and non-fall states, distinguishing them is not straightforward. Some auxiliary parameters are required in order to do so. These parameters can be determined from the basic fact that feature variables tend to undergo drastic changes during the course of falling when compared to the non-falling condition. These changes and the instance at which they happen can be determined using two parameters, collectively called as prediction window parameters, and they can be defined as follows:

- *Thresholds (τ)*: these are the maximum and minimum values which a variable can take during the course of a humanoid's non-falling over condition. In the case of a fall over, the above values are considerably higher or lower respectively, when compared to that of a non-fall case.
- *Fall observation period (f_{obp})*: is a time period within which either the maximum or the minimum values of a certain feature variable exceed the above defined thresholds to be distinguished as a probable fall state.

4-3-1 Selection of Prediction Window Parameters and Their Effects

The aforementioned parameters have to be determined appropriately in order to achieve the desired distinguishing performance with the selected FIV. The selection of suitable parameters and their effects on determining a fall over state are explained here with the evolution of an FIV, given a disturbance to the robot. For instance, in Fig. 4-3, the orientation of a humanoid robot about its y-axis (IMU_{oy}) has been considered for illustration. Figure 4-3a shows the evolution of IMU_{oy} during the course of a humanoid's non-fall, and Fig. 4-3b shows the same during its falling over condition. Figure 4-3c shows how τ and f_{obp} together affect the distinguishing performance.

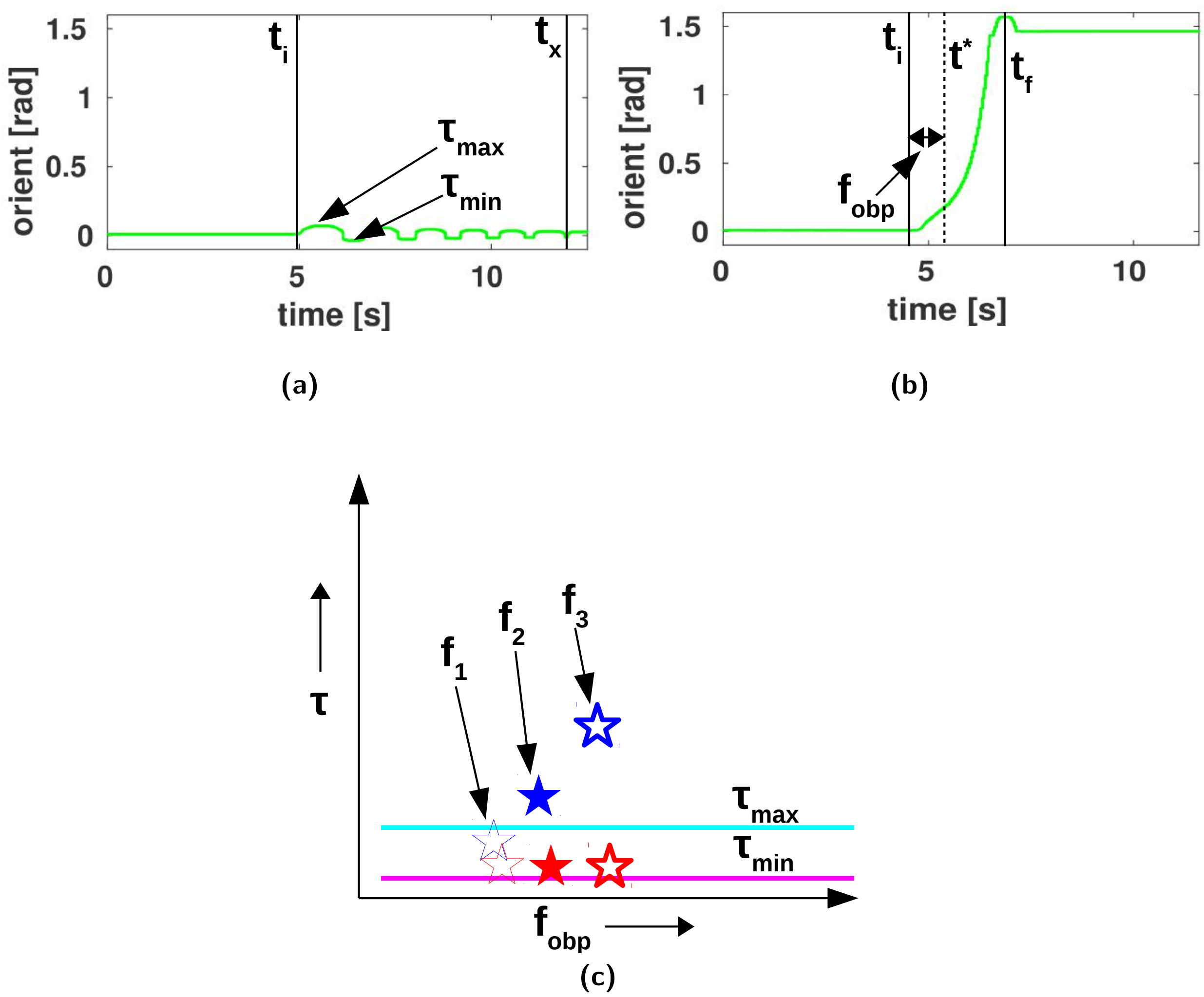


Figure 4-3: (a) Time evolution of a humanoid's orientation about y-axis is plotted against time for a non-fall case. t_i and t_x represent the impact application instant and the end time considered for extracting thresholds, (b) Evolution plot of the same FIV for a fall case is shown. $t^\star$ and t_f are the end time of fall observation period (f_{obp}) and the time at which the robot hits the floor, and (c) Maximum (all blue stars) and minimum (all red stars) values extracted from the fall plot for different f_{obp} are shown along with the maximum (blue line) and minimum (magenta line) thresholds determined from the non-fall plot. f_1, f_2, and f_3 represent set of values (light, shaded and dark stars) obtained with different f_{obp} in the ascending order.

For computing τ, it is necessary to consider certain data range from the instant of impact application (t_i). Ideally, the data can be of any arbitrary length, but it should be sufficient enough to include the maximum, minimum, and the variation of any chosen FIV. In Fig. 4-3a, which plots the time evolution of IMU_{oy}, t_x denotes the end time sample of the data range and it is approximately taken to be the point at which the FIV stabilizes. The period $t_x - t_i$ is considered for the extraction of τ, and it includes the data's variation and extremities.

In the case of f_{obp}, which is given by $t^\star - t_i$ as shown in Fig. 4-3b, $t^\star$ has to be chosen appropriately such that sufficient amount of data is observed to make a reliable decision. If f_{obp} is too small (i.e., $t^\star$ too close to t_i) then the maximum (blue star) and minimum (red star) values (f_1) will be within the non-fall thresholds resulting in an ambiguous condition. If f_{obp} is too big (i.e., $t^\star$ too close to t_f) then the maximum value of fall data is significantly above τ_{max} (f_2), making it easier to predict fall, but probably it is too late. An adequate f_{obp} should be chosen such that it is sufficient enough to avoid f_1 and f_3 condition, and end up with a situation like f_2 where either the maximum (shaded blue stars) or minimum (shaded red stars) value is above their respective threshold to distinguish a fall over state. Also, considering a set of data instead of an instantaneous value provides the following advantages: helps to tackle noisy observations through filtering, takes past data and the variation of a particular FIV into account to make a reliable decision.

4-4 Procedure to Determine Prediction Window Parameters

In the previous subsection, prediction window parameters, their requirements, and their effects on recognizing a fall over state were discussed with an example. In this section, we intend to determine the above parameters in a generic way such that they can be used offline to distinguish between any fall and non-fall state, and later they can be used online to predict the same. It is then necessary to capture a wide range of data to represent the generic evolution of FIV during the course of fall and non-fall over motions. This data can be generated by applying different magnitude of disturbances at varied locations, until the robot falls, and storing the non-fall and fall data separately for further analysis. Figure 4-2a shows the steps involved in the determination of parameters. The following subsections first introduce the experimental setup, data processing, and finally, the determination of parameters from the processed data.

4-4-1 Fall and Non-fall Experiments

Since it is highly risky to carry out both the fall and non-fall experiments with the real robot, and also it is extremely time-consuming to perform a lot of experiments, a simulator setup is used. Figure 4-4 shows the setup, which includes a pushing structure and a humanoid, used to extract the raw data of the variables considered for each feature. The pushing structure exerts impact to the robot in the x-direction with Sliding Ram 1, and one can also change the height of the impact point in the z-direction with Sliding Ram 2, where their motion range can be determined by considering the robot's size. Impact

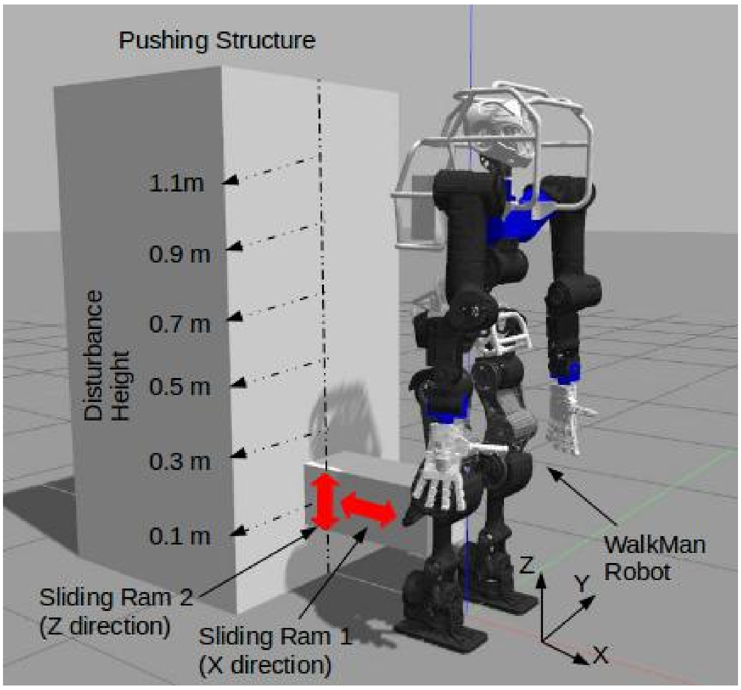


Figure 4-4: Simulator set-up for data collection

exertion at various heights is necessary to observe the dynamic effects, such as the force required for toppling (higher force with increased height and vice versa), and feet slipping, which are in general caused by the non-uniform distribution of robot's mass in the feature variables.

For experiments in this paper, for example, Sliding Ram 1 applies a force in the range of 100 - 3000 N for 30 ms, which is gradually increased until the robot falls. The forces are applied at 6 different heights from 0-1.1 m by Sliding Ram 2. Indeed, in total, 36 experiments were carried out (30 non-fall and 6 fall cases).

4-4-2 Data Processing

The fall and non-fall data gathered from the experiments in the previous subsection are generally noisy, and they are filtered using a Hampel and median filter combination. The Hampel filter removes the outliers [35], and the median filter successively smoothens the signal without distorting its original trend [36].

Figure 4-5 shows the data obtained when a disturbance of 1200 N for 30 ms is exerted on the robot in the forward direction. Filtered data plots for the pushing force and feature variables listed in Table 4-1 are respectively shown in Figs. 4-5a - 4-5f. The impulse is applied at 14.05s, as shown in Fig. 4-5a, and its effect is respectively observed in all the plots. In the CoG plot (Fig. 4-5c), considerable variations are observed in the position, velocity, and acceleration of CoG in the x-direction, due to the impulse applied along that

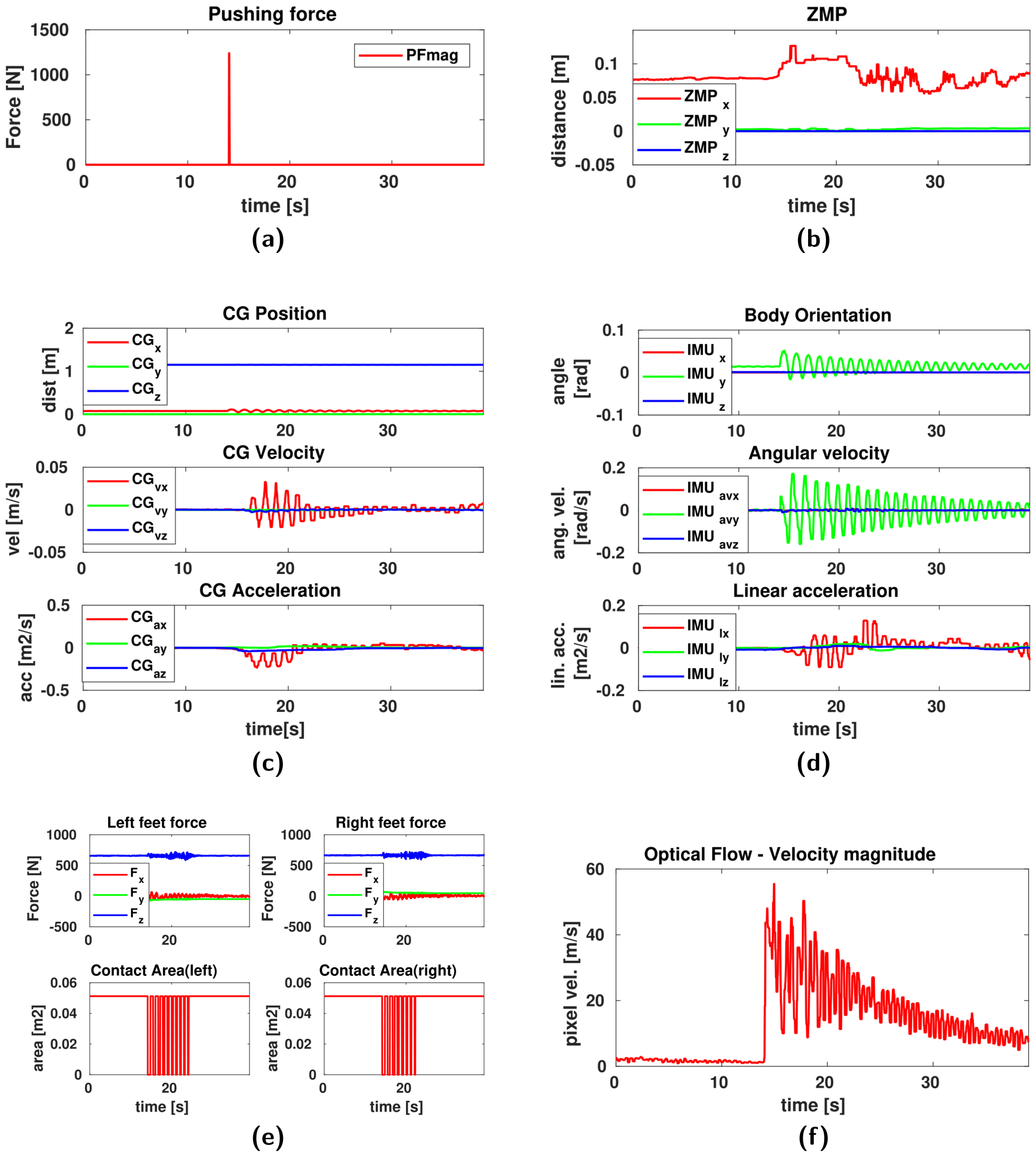


Figure 4-5: Filtered data plot for impulse exerted on the robot at 14.05s at a height of 0.9 m: (a) disturbance force, (b) ZMP, (c) CoG position, velocity and acceleration, (d) IMU orientation, its rate and linear acceleration, (e) feet contact force and area, and (f) optical flow velocity.

direction. Similarly in Fig. 4-5d, variations are observed in the robot's orientation, and its rate along the y-axis, and linear acceleration in the x-direction, while for ZMP (Fig. 4-5b), the difference is observed in its x position. In the contact sensor plot Fig. 4-5e, constant fluctuations are observed due to mild oscillations of the robot caused by the disturbance. In Fig. 4-5f, the optical flow shows a high velocity of the features (features selected from the image), immediately after the application of disturbance, and gradually settles down as the robot stabilizes slowly. Due to the oscillatory motion of the robot, a constant zig-zag flow velocity is observed of the features being tracked.

4-4-3 Determination of Prediction Window Parameters

In this subsection, we address how to determine τ and f_{obp} to distinguish the fall- and non-fall cases with the filtered data obtained from the previous experiments, e.g., 36 experiments are considered as explained in Section 4-4-1.

First, the maximum and minimum threshold of each variable which together forms τ of those variables are determined. As discussed earlier, to determine τ an appropriate data length from the non-fall experimental data has to be considered. The data range can be of any arbitrary length, but it should be adequate enough to capture the extremities and variations of the variables. For instance, in the presented work 1000 time samples from t_i have been considered. The data length taken here was found to be sufficient enough to capture the requisite data characteristics from all 30 non-fall data sets.

In the case of f_{obp}, a certain portion of data from the fall period $f_p = t_f - t_i$ is used for observation, where t_f is the instant of humanoid's complete fall over shown in Fig. 4-3b. The end time sample of f_{obp} can be computed as $t^* = x(t_f - t_i) + t_i$, where x can vary between 0 to 1. An appropriate value is obtained by gradually lowering the amount of fall data considered for observation until the τ determined above were able to distinguish the fall states clearly. For the results presented here, x is taken to be 0.3, which is nothing but 30% of fall period data from each fall experiment.

4-4-4 Parameters Evaluation and FIVs

In this subsection, the parameters obtained above for different feature variables and their ability to distinguish fall over states are presented here in Figs. 4-6, 4-7, and 4-8. Later, based on the results, feature variables which are effective in recognizing a fall state are determined and they are collectively termed as FIV.

Figure 4-6a displays the threshold plot obtained for the CoG feature. The light blue and magenta lines denote the maximum and minimum threshold of each variable, considering the non-fall data, while the dark blue and red stars denote the maximum and minimum values of the variables for each fall case corresponding to their respective disturbance height.

In the figure, the first row represents CoG position (CG_x, CG_y and CG_z), and second row represents CoG velocity (CG_{vx}, CG_{vy} and CG_{vz}). The figure clearly shows that for the fall cases (dark blue stars) CG_x and CG_{vx} is clearly outside the maximum thresholds, indicating that it is falling forward. Similarly, both CG_z and CG_{vz} is below the minimum threshold, suggesting that the robot is falling down. No substantial difference observed in CG_y and CG_{vy} since the robot is disturbed only in the x-direction. Due to the erratic distribution of the acceleration data, it is neglected and not shown here for brevity.

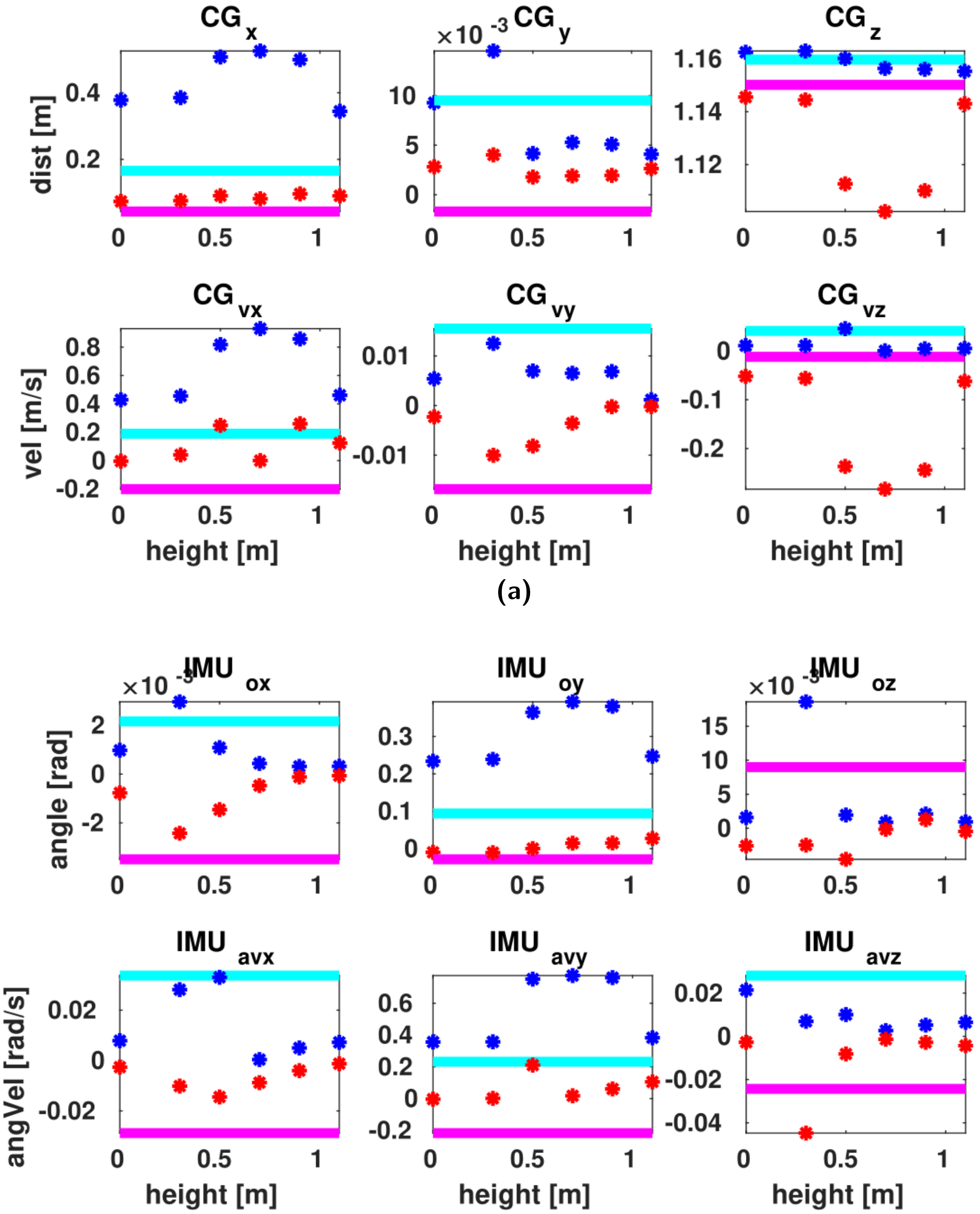
In Fig. 4-6b, the first row represents the orientation in xyz axes (IMU_{ox}, IMU_{oy}, and IMU_{oz}), and second row represents the angular velocity (IMU_{avx}, IMU_{avy} and IMU_{avz}). The linear acceleration is not considered for the same reason as aforementioned. Substantial differences are observed only in IMU_{oy} and IMU_{avy} because the robot is tilting about the y-axis due to the disturbance applied in the x-direction.

Figure 4-7a shows the foot contact force thresholds and it uses the following metrics: force magnitude in the z-direction (1st row) and force gradient (2nd row). In the case of force magnitude, the values of the fall cases were found to be below the min threshold, except for one case. With the force gradient, the values corresponding to the fall cases were consistently negative, and they are found to be considerably lower than the minimum value of the non-fall cases. The negative gradient suggests that the robot is losing foot contact with the ground. In Fig. 4-7b, foot contact area thresholds are obtained by taking the mean of the contact area and its standard deviation. Considering both metrics, it is observed that the corresponding values for the fall cases are lower than the minimum value of the non-fall cases. The lower mean contact area of the fall cases suggest that the robot has point contact predominantly, and the lower standard deviation denotes that the mean contact area is not changing much, reaffirming that the robot is not oscillating and it is going down. The relatively high standard deviation of the mean contact area observed for the non-fall cases is due to the fact that the robot is oscillating between a surface and a point contact considerably, before settling down to a stable surface contact.

In Fig. 4-8a, the maximum and minimum ZMP values for x (ZMP_x) and y (ZMP_y) positions are shown for both the fall and the non-fall cases. Substantial differences were observed in ZMP_x between the maximum value of fall and non-fall cases. No significant differences were seen in the ZMP_y position since the robot was disturbed only in the x-direction. Figure 4-8b compares the optical flow velocity (OPF_{vel}) values extracted for the fall and the non-fall cases. OPF_{vel} assumes a relatively high value in the case of fall when compared to the non-fall cases. A similar trend is observed with the standard deviation of OPF_{vel}, but in less number of fall cases.

Based on the differences observed between the non-fall and fall cases in the plots shown in Figs. 4-6, 4-7, and 4-8 we can determine the effective feature variables to predict the fall. As aforementioned, those variables are defined as FIV summarized in Table 4-2.

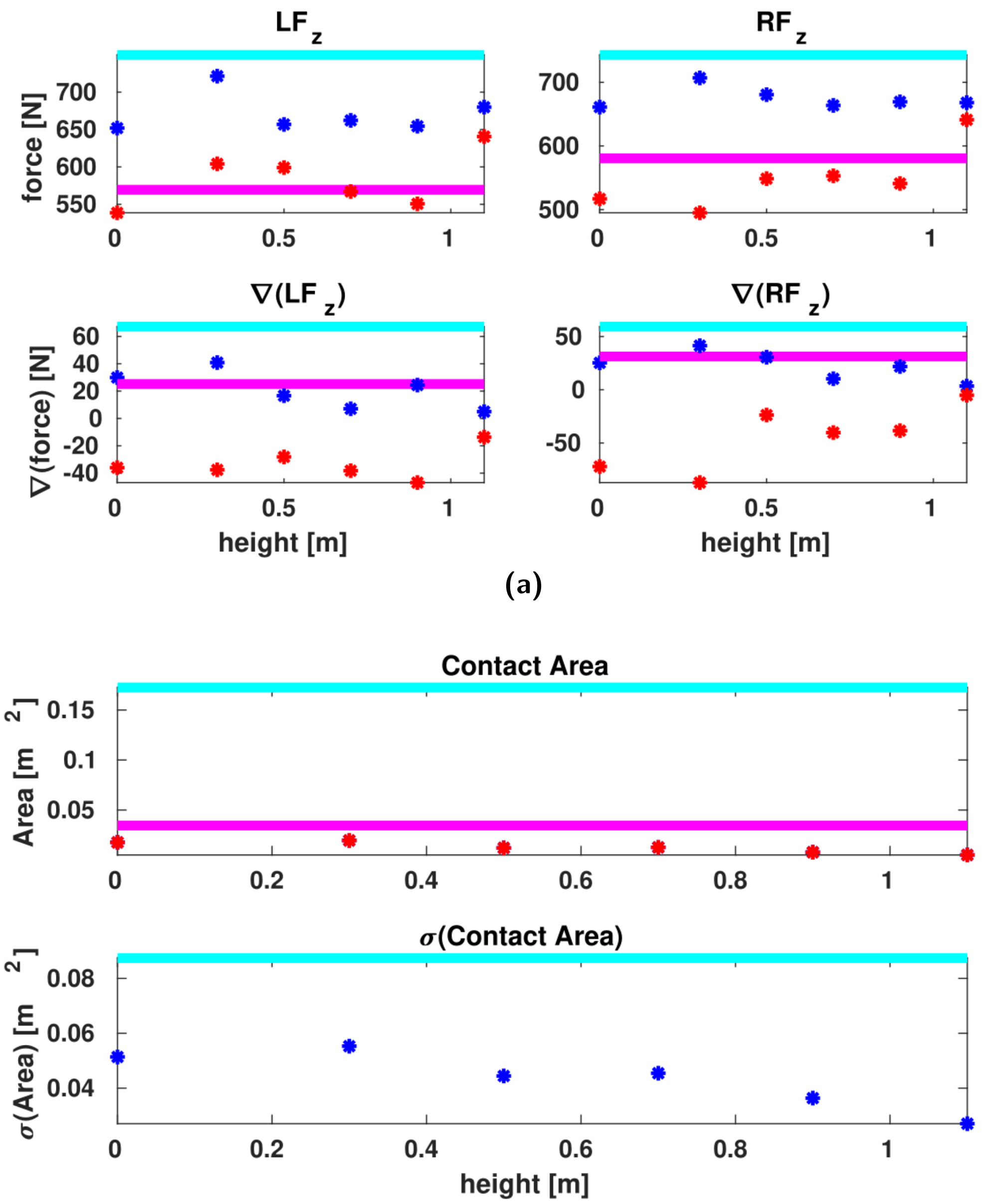
The parameters τ and f_{obp} obtained in this section considers only the data set collected by disturbing the robot from behind with different disturbance magnitudes. In general, a humanoid could be subjected to disturbance in an$y$-direction. Hence, it is necessary for these parameters also to be generic in nature. The disturbance direction has negligible effect on f_{obp}, since it is possible to capture the prime characteristics of a fall state assuming



(a)

(b)

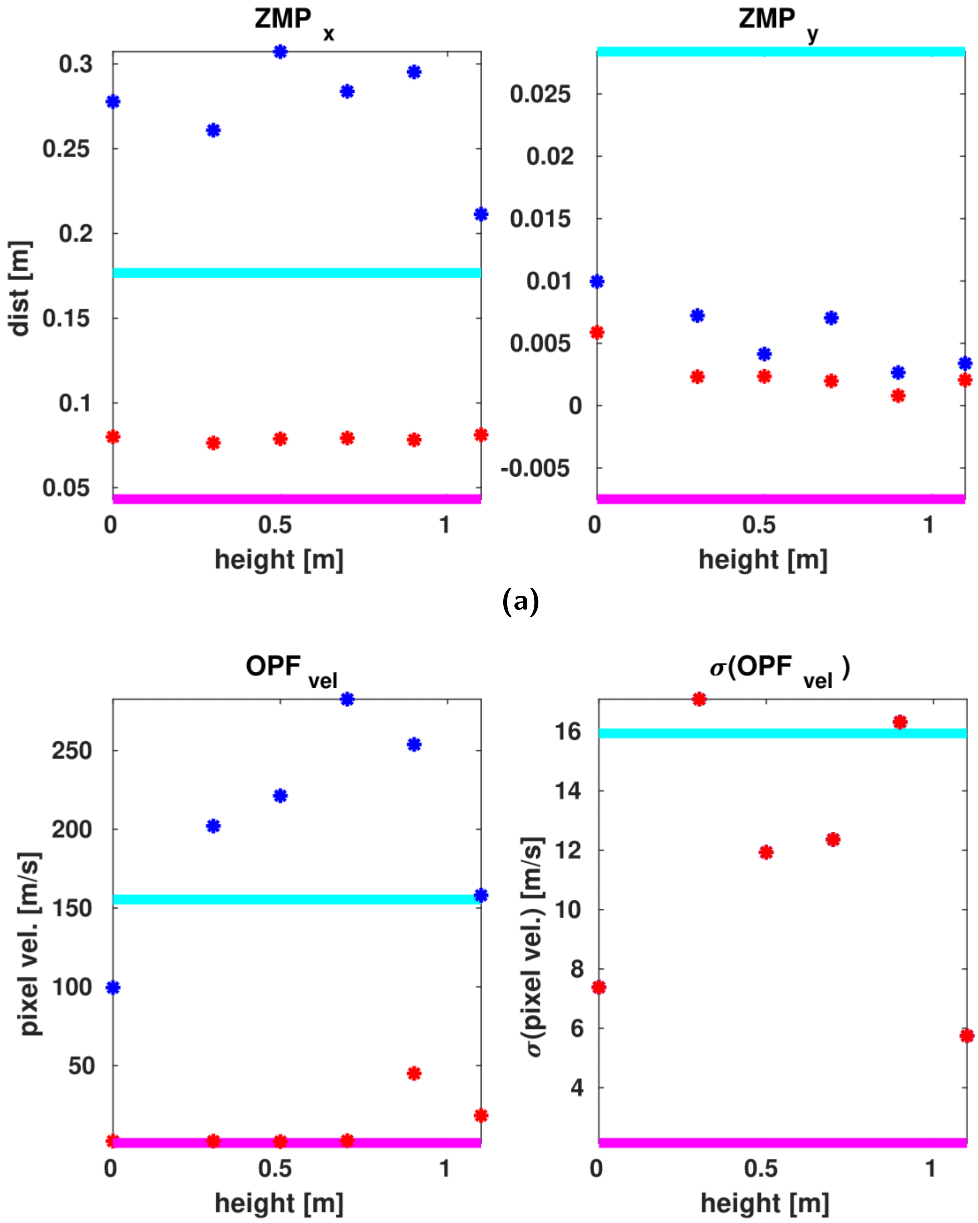
Figure 4-6: The plots show the maximum (light blue line) and minimum (magenta line) thresholds of the non-fall data sets, and the maximum (dark blue) and minimum (dark red) values observed within f_{obp} of each fall data. It also shows how different feature variables classify the fall state and they are plotted in the following order: (a) CoG, and (b) IMU.



(b)

Figure 4-7: The plots show the maximum (light blue line) and minimum (magenta line) thresholds of the non-fall data sets, and the maximum (dark blue) and minimum (dark red) values observed within f_{obp} of each fall data. It also shows how different feature variables classify the fall state and they are plotted in the following order: (a) FPS(force), and (b) FPS(area).

a fairly optimal value. But with τ, both the direction and magnitude of the time evolution of each FIV changes. The following section proposes a method to generalize τ.



(a)

(b)

Figure 4-8: The plots show the maximum (light blue line) and minimum (magenta line) thresholds of the non-fall data sets, and the maximum (dark blue) and minimum (dark red) values observed within f_{obp} of each fall data. It also shows how different feature variables classify the fall state and they are plotted in the following order: (a) ZMP, and (b) OPF.

4-5 Online Threshold Interpolation

The thresholds extracted in the previous section are only applicable for the disturbance applied in the positive x-direction (Scenario 1 in Fig. 4-9). However, in real situations,

Table 4-2: FIV used for the proposed fall prediction.

Features	FIV
CoG	CG_x, CG_y, CG_{vx} and CG_{vy}
IMU	IMU_{ox}, IMU_{oy}, IMU_{avx} and IMU_{avy}
ZMP	ZMP_x and ZMP_y
FC	LF_z, RF_z, $\nabla(LF_z)$, $\nabla(RF_z)$, F_{ca} and $\sigma(F_{ca})$
OPF	OPF_{vel} and $\sigma(OPF_{vel})$

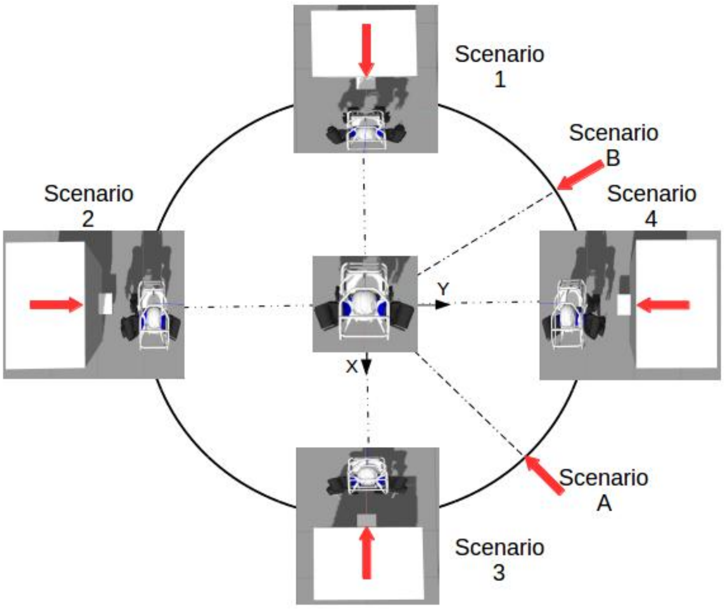


Figure 4-9: Four major scenarios for manual threshold extraction. Scenarios A and B are intermediate ones.

the disturbance can be applied in any direction, and it is very difficult and tiresome to manually extract the thresholds for all such cases. Hence, for four major disturbance directions thresholds, τ_{major}, were extracted as shown in Fig. 4-9 using the extraction technique described in Section 4-4-3. These directions were selected based on the impact it has on the variables due to the nature of mass distribution, impact point location, and variation in the support polygon.

With the thresholds extracted for 4 major scenarios, an online threshold interpolation technique is proposed to handle other intermediate scenarios like A and B shown in Fig. 4-9. This is done as shown in Fig. 4-10, using the CoG velocity, where a vector $\mathbf{v}_{CoG}$ is constructed, and the angles (a_1,a_2,a_3 and a_4) between $\mathbf{v}_{CoG}$ and all the major disturbance

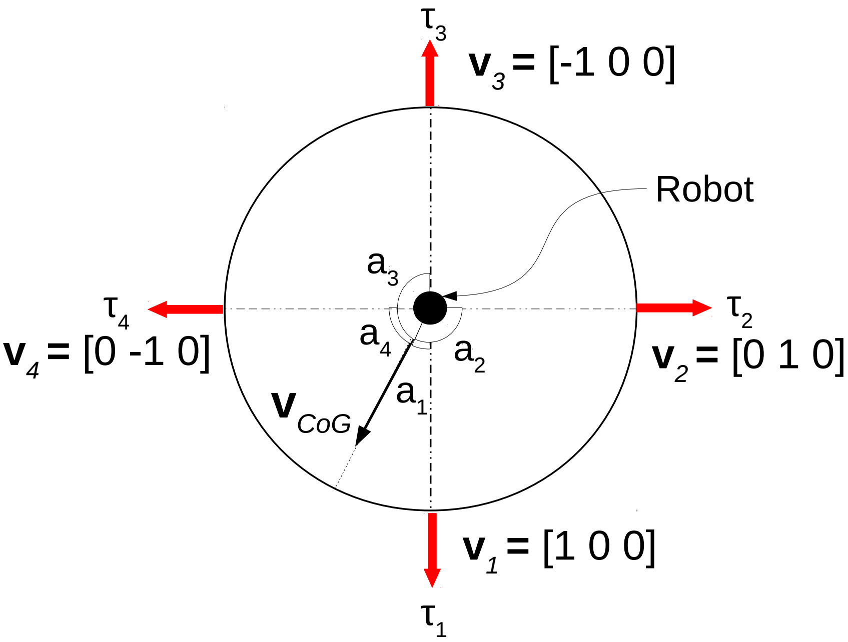


Figure 4-10: Online threshold Interpolation.

direction vectors ($\mathbf{v}_1$,$\mathbf{v}_2$,$\mathbf{v}_3$ and $\mathbf{v}_4$) are computed as follows:

$$a_i = \tan^{-1}\left(\frac{\mathbf{v}_{CoG} \times \mathbf{v}_i}{\mathbf{v}_{CoG} \cdot \mathbf{v}_i}\right) \tag{4-1}$$

By comparing all four angles, the quadrant in which $\mathbf{v}_{CoG}$ lies is determined, and the ratio of closeness, r_1 and r_2, to the quadrant formed by the two threshold vectors are then computed as $r_1 = \frac{a_2}{a_1+a_2}$ and $r_2 = \frac{a_1}{a_1+a_2}$. Using the closeness ratio, the new threshold ($\tau^\star$) can be calculated instantaneously as

$$\tau^\star = r_1 \cdot \tau_1 + r_2 \cdot \tau_2. \tag{4-2}$$

In this way, the thresholds for disturbance applied to the robot in any other direction can be instantaneously computed with only four sets of thresholds.

4-6 Fall Indicator

To predict a fall using the features, a continuous variable called fall indicator (ϕ) which can vary from 0-1 is used. A value of 0 indicates the lowest chance of the robot to fall over, while a value of 1 indicates the highest chance of its fall. The ϕ value for each feature is computed as follows:

$$\phi = \frac{1}{N}\sum_{i=1}^{N}\phi_i, \tag{4-3}$$

where ϕ_i denotes the fall indicator value computed using the i^{th} variable of a particular feature, and N is the total number of variables. ϕ_i values are the degree of closeness to their respective upper and lower thresholds, and it is computed using two different types of cost functions as given below:

- Type 1: it is linear, and it measures the degree of closeness of a given value, $X^\star$, to its upper and lower thresholds, which are X^u and X^l respectively. The function is defined as follows:

$$\phi_i = \begin{cases} \left| \dfrac{X^\star - \left(\frac{X^u + X^l}{2}\right)}{\left(\frac{X^u - X^l}{2}\right)} \right| & \qquad\qquad\qquad\qquad \text{(4-4)} \\ 1, \qquad\qquad\qquad \text{if } \phi_i > 1. & \qquad\qquad\qquad\qquad \text{(4-5)} \end{cases}$$

- Type 2: this is used for quantifying exponentially how far $X^\star$ is from X^u and X^l, and it is done as shown below:

$$\phi_i = \begin{cases} e^{\left(\frac{X^\star - X^u}{X^u}\right)}, & \qquad\qquad\qquad\qquad \text{(4-6)} \\ e^{\left(\frac{-X^\star}{X^l}\right)}, & \qquad\qquad\qquad\qquad \text{(4-7)} \\ 1, \qquad\qquad\qquad \text{if } \phi_i > 1. & \qquad\qquad\qquad\qquad \text{(4-8)} \end{cases}$$

  For computing the upper and lower threshold closeness (4-6) and (4-7) are used respectively.

For the variables associated with the features CoG, IMU, and ZMP, (4-4) and (4-5) are used for computing ϕ_i, while exponential functions are used for FC and OPF. The usage of latter is to account for the abrupt variation observed with the features of FC and OPF, in contrast to the feature variables of CoG, IMU, and ZMP. In particular, (7) is used for the variables of FC, since it is evident from Figs. 4-7a and 4-7b in Section 4-4-3 that the variables for fall cases are always less than the lower thresholds irrespective of the disturbance direction. A similar trend is observed with OPF variables, but they are constantly above the upper thresholds for fall cases as shown in Fig. 4-8b. The above trends are never observed with the variables of CoG, IMU, and ZMP since their variations are direction specific.

Figure 4-2b shows the robot's fall prediction process with feature-based fall indicators. With the robot subjected to disturbance, the data (i.e., feature variables) are gathered, filtered, and then sent to the online threshold interpolation module, if the thresholds have not been set already. Based on the disturbance direction, the threshold interpolation module computes the respective thresholds for each feature based variables. Upon computing a threshold, filtered data are passed on to the process block ♣ illustrated in Fig. 4-2b, wherein ϕ_i is computed in proportion to the variable's distance from its respective thresholds. In the end, ϕ for a particular feature is computed using the set of ϕ_i as given in (4-3). Here, ϕ is computed considering a single feature, whereas our proposed method is based on multiple features, which is discussed in Chapter 5.

4-7 Fall Monitoring

Fall monitoring is the conditional evaluation done for the fall over of the humanoid, where F_{pred} is to be evaluated as 1 or 0. The condition considered here for the evaluation of F_{pred} is as follows:

$$F_{pred} = \begin{cases} 1, & (\phi \geq \phi_{critical}) \text{ and } (F_{mon} > T_m) \\ 0, & \text{otherwise,} \end{cases} \tag{4-9}$$

where $\phi_{critical}$ is the critical fall indicator value set to a constant number 0 to 1 which indicates a high chance of falling over, e.g., $\phi_{critical} = 0.8$, F_{mon} is the fall monitoring counter, and T_m is its limit which is modified online based on the impulse subjected to the robot and hence called the impulse adaptive counter limit. The counter here denotes the number of times the condition $\phi \geq \phi_{critical}$ remains satisfied continuously. The following subsections discuss the determination of T_m and $\phi_{critical}$.

4-7-1 Impulse Adaptive Counter Limit

In order to make the above prediction process more reliable and agile, T_m was introduced in (4-9). Though it is difficult to make the prediction method 100% reliable and agile, it is possible to achieve a balance between the two criteria. The above characteristics of the prediction method are determined by the evolution of features, which in turn are dependent on the disturbance given to the system.

Considering the system as a linear inverted pendulum model, the disturbances exerted on the system can be generalized to an impulse I and it is written as

$$I = F\Delta t = m\Delta v, \tag{4-10}$$

where F is the disturbance force applied for Δt duration, m is the mass of the system, and Δv is the change in velocity. Since m is constant, the impulse effect is directly visualized in $\Delta v = v^+ - v^-$, where v^+ and v^- are the velocity of the system before and after the application of the impulse. Taking $v^- = 0$, Δv can be written as

$$\Delta v = v^+ = r\omega^+ \tag{4-11}$$

where, r is the distance between the tipping point and the robot's CoG, and ω^+ is the tipping velocity after the application of impulse. The above computed v^+ is normalized to

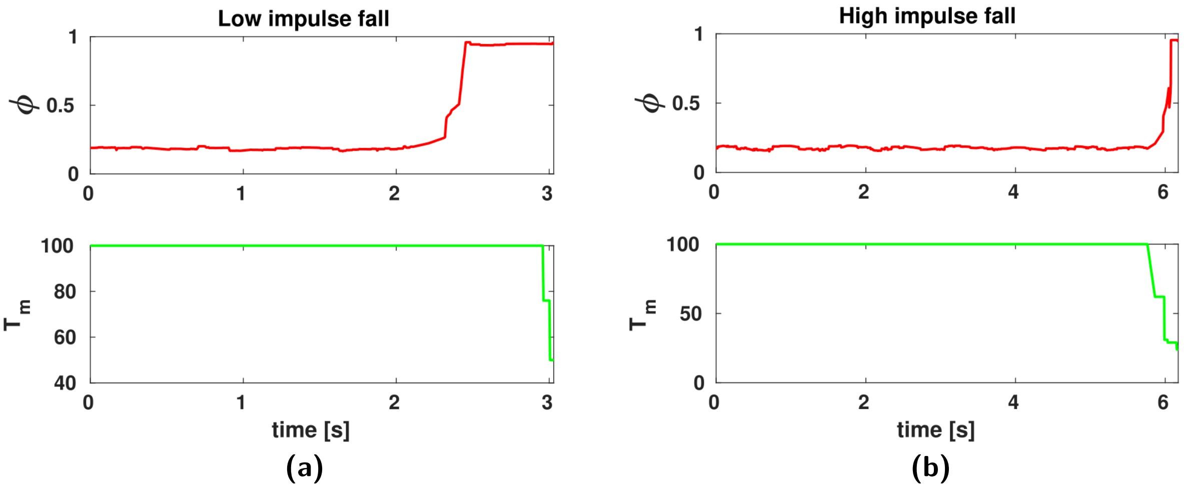


Figure 4-11: Evolution of T_m and ϕ shown for different magnitude of impulses: (a) a low impulse fall modifies the value of T_m online from 100 to 50, and (b) a high impulse fall results in T_m to change from 100 to 24.

1 ($\hat{v}^+$), using the minimum and maximum post impulse velocities v^+_{min} and v^+_{max} as follows:

$$\hat{v}^+ = \frac{v^+ - \frac{v^+_{max} - v^+_{min}}{2}}{\frac{v^+_{max} - v^+_{min}}{2}}. \tag{4-12}$$

v^+_{min} is computed as shown in (4-11) with ω^+_{min} computed using the IMU angular velocity thresholds, IMU_{avx} and IMU_{avy}. For the computation of v^+_{max}, ω^+_{max} which can result in a minimum fall over time[1] is used, and this value is approximately determined by simulating a linear inverted pendulum model.

T_m is modified online by making use of the above computed $\hat{v}^+$ inside a non-linear function as shown below:

$$T_m = e^{(-\alpha \times \hat{v}) \times T_{mon}}, \tag{4-13}$$

where α is a positive integer used to vary the growth rate of the exponential function, and T_{mon} is the maximum number of time steps to monitor (100 by default). For lower impulse, T_m takes the default value, and as the impulse is increased, T_m decreases exponentially. This variation lets the prediction method react slow in order to prune robot's swaying motion during low impulse disturbances, which in turn reduces the *false positive* (FP) case. Whereas, for high impulse disturbance cases, the proposed function makes the method more agile to reduce the instances of late fall prediction (LFPr).

[1] In this paper, 1.0s is taken as the minimum fall over time for two reasons: 1) the maximum achievable fall over time with minimum prediction failures has been reported as approximately 1.0s,[24] and 2) as the rate of fall over time decreases, the efficiency of some of the prominent controlled fall strategies reduces considerably [37].

Figure 4-11 shows the evolution of T_m and ϕ given a low and high impulse disturbance to the robot. In the case of a low impulse (24 Ns), shown in Fig. 4-11a, the evolution of ϕ (top plot) is relatively slow due to the robot's slow fall movement. This makes the proposed technique to maintain a higher T_m, and later, reduces to 50 as the fall movement gains speed. While in the case of high impulse (45 Ns), the proposed technique senses the robot's relatively fast fall movement and modifies the value of T_m swiftly to 24 from 100, as shown in Fig. 4-11b.

4-7-2 Critical Fall Indicator Value

An appropriate $\phi_{critical}$ is obtained after numerically evaluating 10 different values (0-1 incremented by 0.1) by subjecting to 36 different disturbances, as shown in Fig. 6-1b. The results are shown in Fig. 4-12 and Fig. 4-13, where Fig. 4-12 shows the normalized prediction time obtained for different cases by assuming various $\phi_{critical}$ values. The blue color in the figure represents prediction failure either due to false positive or negative alarms. Figure 4-13 shows the average prediction time (top) and success rate (bottom) plotted against $\phi_{critical}$. The evaluation is based on the success rate and the time robot takes to fall from the instant of prediction. For lower $\phi_{critical}$ values (0.1-0.4), high fall time is observed but very less success rate and vice versa for extreme values (0.9-1). A balance between the two is seen in the mid-range (0.5-0.8), with 0.8 excelling in both criteria and hence it is chosen here.

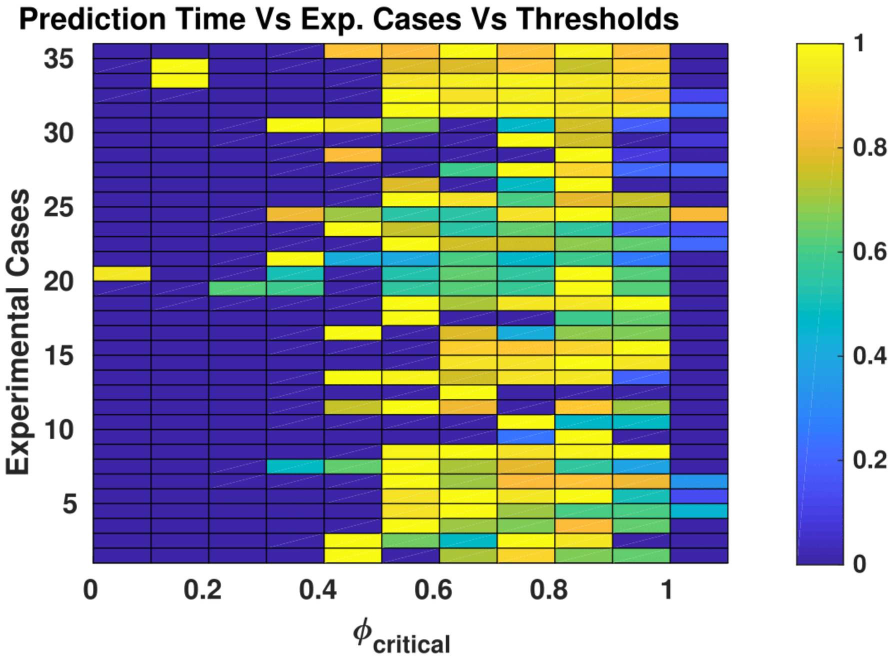


Figure 4-12: Prediction time for 36 different test cases with various critical fall indicator ($\phi_{critical}$) values.

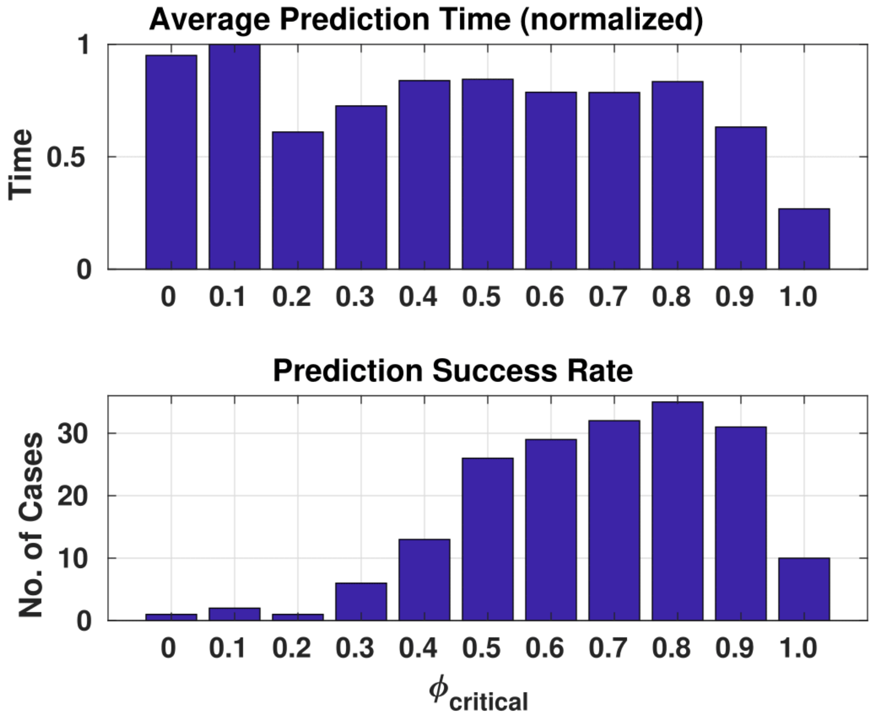


Figure 4-13: Average prediction time and success rate plotted against critical fall indicator ($\phi_{critical}$) values.

Chapter 5

Fall Indicator Estimation Techniques Using Multiple Features

As discussed in Section 4-6, ϕ can be computed based on a single feature, but with multiple features it is possible to predict fall in a reliable and agile manner, and also handle different terrains even in the presence of noise. The above merits are mainly due to the individual characteristics of each feature, which cannot be availed with ϕ based on a single feature. Hence, in this section, we propose two different methods: 1) Kalman filter-based (KF) and 2) performance-based (PF) techniques to estimate the optimal value of ϕ. While KF estimates an optimal value ϕ_{KF}, PF gives an empirical value ϕ_{PF}. The performance of each feature individually and their fused version on various terrains, and also its behavior in the presence of noise are extensively analyzed and compared in Chapter 6.

5-1 Kalman Filter Based Technique (KF)

Kalman filter, a fusion technique, is used here for combining the individual ϕ obtained from each feature to cope up with the different level of noise observed in each sensor. Figure 5-1 shows an overview of the fusion using Kalman filter, which includes a prediction and a correction section. In the prediction section, the fall indicator is predicted using the variables of a generic and reliable feature, which is taken to be FC. The current state of the fall indicator ($\hat{\mathbf{x}}_k^- \in \mathbb{R}^n$, $n = 6$) and the error covariance matrix ($\mathbf{P}_k^- \in \mathbb{R}^{n \times n}$) are estimated, to obtain an apriori estimate for the next time step. The equations involved are shown below.

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} \tag{5-1}$$

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q} \tag{5-2}$$

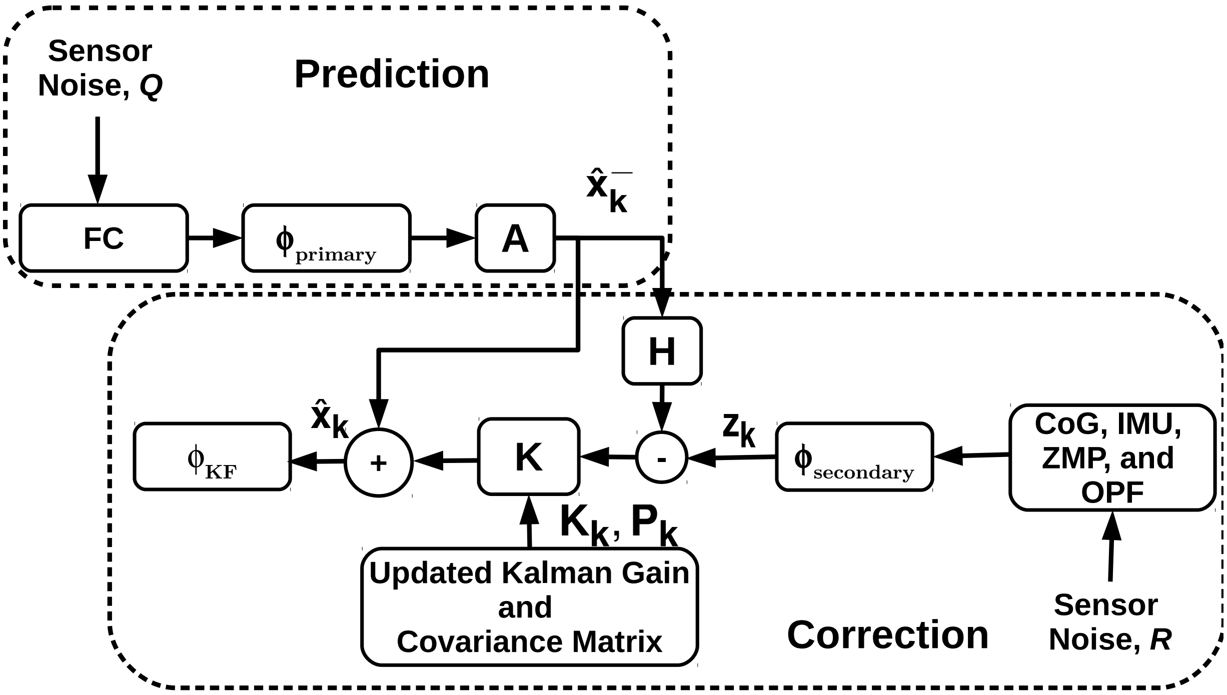


Figure 5-1: An overview of sensor fusion using Kalman Filter.

$\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{B} \in \mathbb{R}^{n \times m}$ represent the system matrix to predict the next state, $\mathbf{u} \in \mathbb{R}^m$ represents the input, $\boldsymbol{\phi}_{primary} = \phi_{\text{FC}} \in \mathbb{R}^n$ is a vector of fall indicator values computed from the primary sensor feature, i.e., foot contact (FC), and $\mathbf{Q} \in \mathbb{R}^{n \times n}$ denotes the noise in the selected feature variables. In (5-1), $\hat{\mathbf{x}}_{k-1}$ is nothing but $\boldsymbol{\phi}_{primary}$. Since $\boldsymbol{\phi}_{primary}$ computed from the sensor readings changes instantaneously (non-linear) when subjected to disturbances, $\mathbf{A}$ is taken to be an identity matrix and the second term in (5-1) is taken as 0.

The apriori estimate $\hat{\mathbf{x}}_k^-$ is corrected using the measurement obtained from the other features $(\mathbf{z}_k)$ to obtain a posteriori estimate $(\hat{\mathbf{x}}_k)$ in the correction section. The Kalman gain $(\mathbf{K})$ and the error covariance matrix $(\mathbf{P}_k)$ are also updated here. Equations (5-3), (5-4), and (5-5) explain the above process.

$$\mathbf{K}_k^- = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1} \tag{5-3}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H} \hat{\mathbf{x}}_k^-) \tag{5-4}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^- \tag{5-5}$$

$\mathbf{z}_k \in \mathbb{R}^m$ is the cumulative ϕ values calculated with the remaining sensor features, which is denoted by $\boldsymbol{\phi}_{secondary} = [\phi_{CoG}\, \phi_{IMU}\, \phi_{ZMP}\, \phi_{OPF}] \in \mathbb{R}^m$. Moreover, $\mathbf{H} \in \mathbb{R}^{m \times n}$ denotes the measurement matrix with its elements assigned as $1/n$ to give equal weighting to all the elements of $\hat{\mathbf{x}}_k^-$ while computing the predicted measurement vector $(\mathbf{H}\hat{\mathbf{x}}_k^-)$. Finally, the cumulative value of $\hat{\mathbf{x}}_k$ yields the optimal fall indicator value, ϕ_{KF}.

5-2 Performance Based Technique (PF)

Another technique of fusing the multiple features is based on their individual performances, to maximize the overall prediction performance of the features together. To fuse the features based on each one's prediction performance, it is necessary to use some metrics to evaluate. The following subsection defines a set of metrics to evaluate the prediction performance and a scalar score to quantify it.

5-2-1 Performance Metrics and Performance Score

The prediction performance of each feature is evaluated by computing individual ϕ in block ♣, as shown in Fig. 4-2b. Four criteria were used to evaluate the feature's performance as follows:

1) *Early fall prediction* (**FPr**) – $(IMU_{ox}||IMU_{oy} \leq \theta^{\star}$ rad)

2) *Late fall prediction* (**LFPr**) – $(IMU_{ox}||IMU_{oy} > \theta^{\star}$ rad)

3) *False positive* (**FP**) – a false alarm for the robot fall

4) *False negative* (**FN**) – failure to detect the fall.

The first two criteria (FPr and LFPr) are associated with successful prediction, while the last two represent failure in prediction. $\theta^{\star}$ is the orientation limit considered to distinguish between FPr and LFPr, and it is set to an angle which yields a higher fall over time, for instance, 0.25 rad is taken here which yields a fall over time greater than or equal to 1.0 s.

In order to make the evaluation intuitive and also easier for comparison, the performance is quantified by introducing a scalar value called *Prediction Performance Score*[1] χ, and it is computed as follows:

$$\chi = (w_{FPr} \times n_{FPr} \times T_{FPr}) + (w_{LFPr} \times n_{LFPr} \times T_{LFPr}), \tag{5-6}$$

where n_{FPr} and n_{LFPr} denote the number of FPr and LFPr cases, while w_{FPr} and w_{LFPr} are the weights assigned to those cases. $T_{\bullet}$ is the time available for the robot to fall from the instant of prediction, calculated by subtracting the prediction time T_{Pred} from the floor hit time T_{Fall}. Accordingly, T_{FPr} and T_{LFPr} represents the fall over time of the early and

[1] Unlike classification, time is an important assessment criteria in fall prediction, which cannot be accounted for with the conventional classifier evaluation criteria (precise and recall), hence χ is introduced.

late fall prediction cases. Ideally, it is preferable to predict the fall as early as possible (i.e., FPr), however, it is unlikely to achieve FPr in all instances. Thus, to assign higher score for features which reported more FPr cases than LFPr, and also to avoid the situation wherein a feature gets the least score due to its predominant LFPr nature, w_{FPr} is taken to be slightly higher than w_{LFPr} in (5-6), for instance, here 0.6 and 0.4 is considered in this work respectively.

5-2-2 Fusion of Multiple Features Based on χ

The sensors are fused based on χ computed for each feature using the performance criteria discussed above. The scores computed for each feature using (5-6) are summed up to form a cumulative total, and the individual scores are normalized with respect to the total, resulting in the weights used later for fusion. For instance, the weight for center of gravity (CoG) feature can be computed as follows:

$$w_{\text{CoG}} = \frac{\chi_{CoG}}{\chi_{CoG} + \chi_{IMU} + \chi_{ZMP} + \chi_{FC} + \chi_{OPF}}, \tag{5-7}$$

where $w_{\bullet}$ and $\chi_{\bullet}$ indicates the weight and performance score computed for feature $\bullet$.

Using the individual weights (w) and ϕ computed for each feature, ϕ_{PF} can be computed as follows:

$$\begin{aligned} \phi_{\text{PF}} =& w_{CoG} \cdot \phi_{CoG} + w_{IMU} \cdot \phi_{IMU} \\ &+ w_{ZMP} \cdot \phi_{ZMP} + w_{FC} \cdot \phi_{FC} + w_{OF} \cdot \phi_{OF}, \end{aligned} \tag{5-8}$$

where $\phi_{\bullet}$ denotes the fall indicator value of each feature.

Chapter 6

Numerical Evaluation

The prediction performance of each individual feature and the above proposed multiple features based estimation techniques are evaluated here by performing the prediction process as shown in Fig. 4-2, in a simulation setup. Depending upon the estimation technique or the feature considered, the ϕ value is computed accordingly in the process block ♣ shown in Fig. 4-2b. In the real environment, the disturbance could be anything such as external force, rough terrains, low stiffness ground, etc. Some of the aforementioned cases have been considered here for evaluation and their results are discussed below.

6-1 Flat Terrain

Figure 6-1 depicts the simulation setup considered for the evaluation of features on a flat terrain. A brick wall is used for extracting feature points and generating 3-dimensional (3D) optical flow, and a two degrees of freedom (DoF) pushing mechanism is used for exerting disturbances. Figure 6-1b shows the 36 different cases of pushing, in which the disturbance direction, its magnitude, and application height are varied.

6-1-1 Performance Comparison of Individual Features

In total 36 different disturbance cases were evaluated for each feature independently, and their performances are compared using the χ in (5-6). The simulation setup considered for the evaluation of flat terrain and the disturbance directions are shown in Fig. 6-1a and Fig. 6-1b respectively.

Figure 6-2a presents the overall score obtained for each feature in a descending order. For comparison, we have also included the conventional instability indicator, capture point

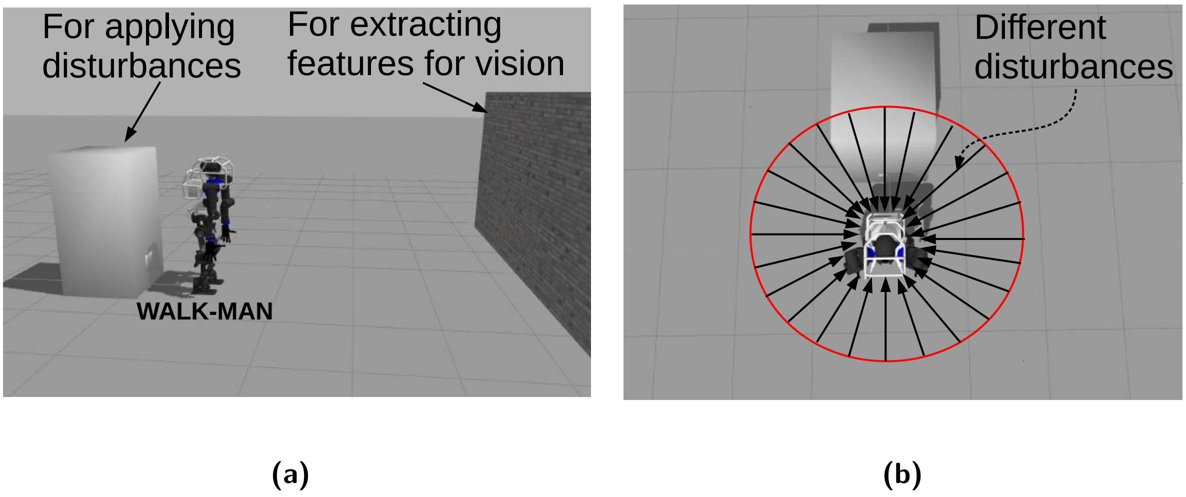


Figure 6-1: Flat terrain simulation setup: (a) Simulation environment along with the components, and (b) Disturbances exerted on the robot in different directions.

(CP), and an updated version of it, which exploits the foot pressure sensor (FPS) called capture point with foot contact (CPF). The three best features for predicting fall of a humanoid robot on a flat terrain are inertial measurement unit (IMU), center of gravity (CoG), and optical flow (OPF), followed by CP, foot contact (FC), CPF, and zero moment point (ZMP). Less difference in performance is observed among the best features, while CP has lower score due to high LFPr(19/36), and FC performs slightly lower due to inherent simulator contact noise and poor handling of swaying motion along the side edges of the feet, which reflects in LFPr(7/36) and FP(7/36). ZMP(LFPr - 13/36, FP - 9/36) has the lowest score due to its high sensitivity to the acceleration of center of mass (CoM).
One of the main advantage of vision is its robustness to noise, and this is shown by adding a zero mean white noise ($\mu = 0.0$, $\sigma = 0.025$) to the sensors and repeating all the scenarios, which alter the three best features relative performance as follows: OPF(12)[1] >IMU(11) $\gg$ CoG(1.512). Though adding noise significantly degrades the performance of all the features, OPF and IMU still maintain a moderate performance level. The above result also supports the notion to consider multiple features during prediction to make reliable and robust decisions under different conditions, which is presented in the following section.

6-1-2 Performance Comparison of Multiple Features

The performance-based (PF) technique described in Section 5-2, and the Kalman filter-based (KF) technique in Section 5-1 are compared using χ as shown in Fig. 6-2b. In the

[1] The positive number in the bracket denotes the performance score χ.

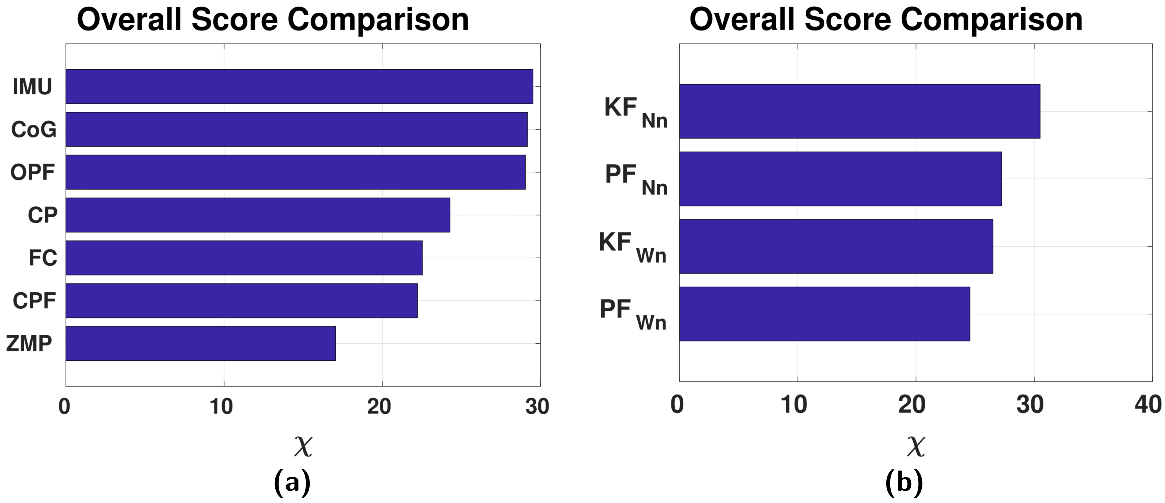


Figure 6-2: Flat terrain prediction performance: (a) single feature-based (without adding noise), and (b) multiple features-based methods (with and without adding noise).

figure, $\mathrm{KF}_{\mathrm{Nn}}$ and $\mathrm{PF}_{\mathrm{Nn}}$ represent the value of χ obtained for KF and PF respectively, with no external noise added to the sensors. The χ values of KF and PF obtained with the addition of external noises are denoted by $\mathrm{KF}_{\mathrm{Wn}}$ and $\mathrm{PF}_{\mathrm{Wn}}$ respectively. The effect of adding external noise to each individual sensor on the proposed techniques is also evaluated here.

In particular, external zero mean noise variance of 0.01, 0.04, and 0.04 were added to the existing[2] values of CoG, IMU, and OPF respectively. We can infer that in both noise free and noisy case, KF performs better than PF, and adding noise reduces the performance of both KF and PF. It can also be seen that χ of KF_{Nn} is better than IMU as found in Fig. 6-2a. This clearly shows that KF outperforms the best individual feature performance, and also maintains a reasonably good level of predictability in the event of noise added to the sensors.

6-2 Uneven Terrain

An uneven terrain is simulated with the set up shown in Fig. 6-3. The set up includes a fixed and a movable platform (in z-direction), the latter covers some square blocks placed randomly on top of the fixed platform, which represent the rough terrain. Since the robot is considered to be static here, it is brought into contact with the rough terrain by suddenly dropping the moving platform. The height (5-25 mm) and the position of the square blocks are varied to generate in total 15 different cases for performance evaluation.

[2] Noiseless case: $P_{k-1} = eye(6,6)$, $R = [0.0739, 0.0664, 0.0742, 0.0696]$, $Q = [0.0578, 0.0935, 0.0997, 0.0480, 0.0891, 0.0926]$.

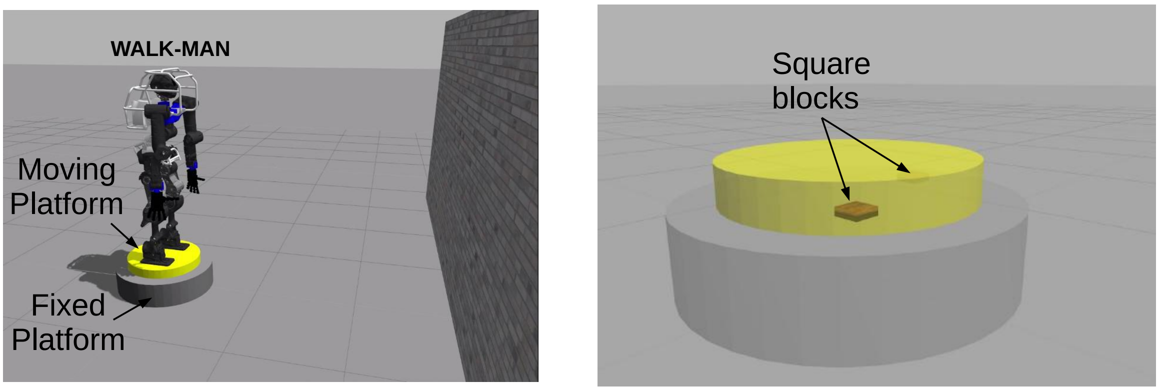


Figure 6-3: Simulator setup for the uneven terrain.

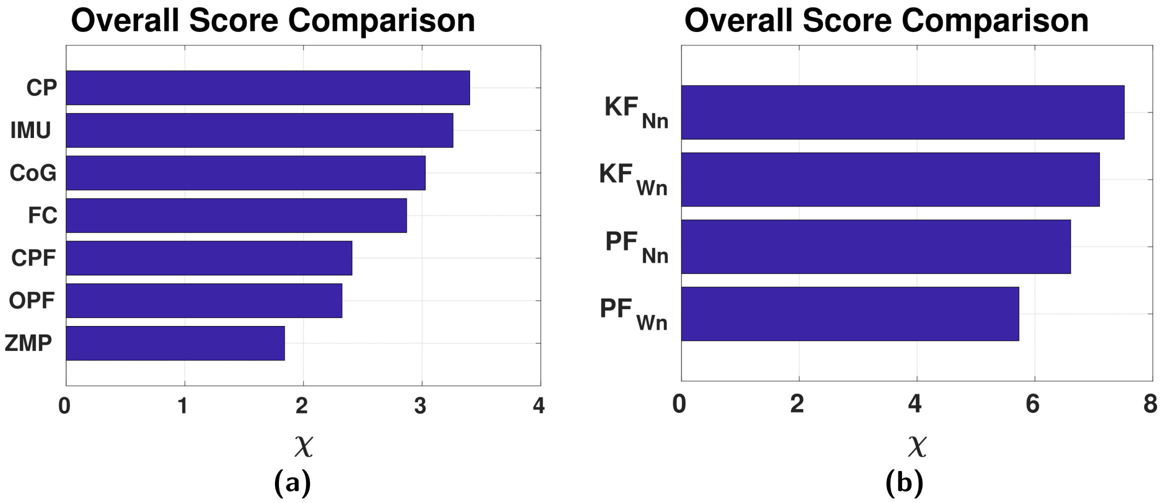


Figure 6-4: Uneven terrain prediction performance: (a) single feature-based (without adding noise), and (b) multiple features-based methods (with and without adding noise).

6-2-1 Performance Comparison of Individual Features

The χ shown in Fig. 6-4a, clearly depicts that features IMU, CoG, and FC perform better than the others. It can also be observed that CP (based on CoG feature) method performs much better than all the individual features. In a noiseless situation, variables of CoG and IMU can be better estimated, explaining their high score. In particular, IMU performs better than CoG which is probably due to the approximate model (simple pendulum) used to compute the latter's variables. FC has slightly lower score due to the contact noise, and OPF performance is low because of high LFPr (13/15) due to the relatively smooth fall motion in some cases, which reflects late in the optical flow since it is based on velocity.

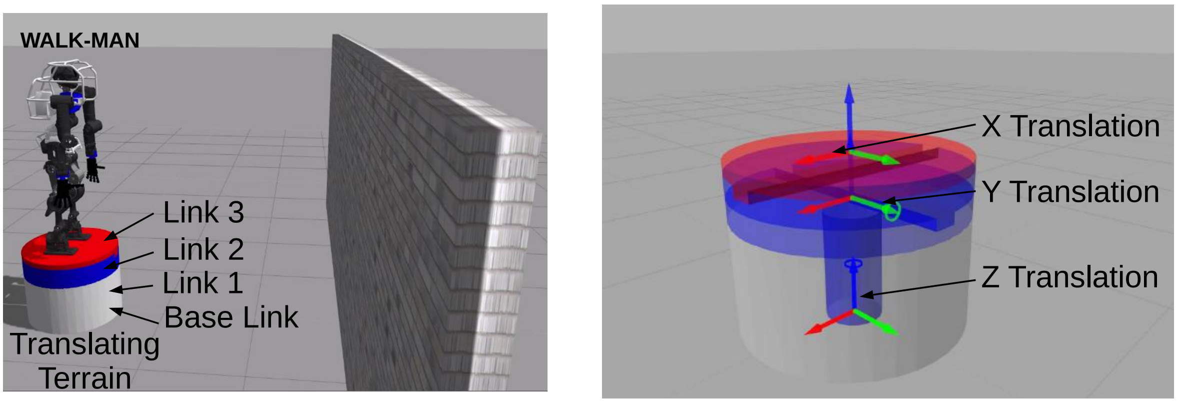


Figure 6-5: Simulator setup for the moving terrain.

6-2-2 Performance Comparison of Multiple Features

Similar to the flat terrain case, uneven terrain (Fig. 6-4b) also shows that KF performs better than PF, both in the presence and absence of noise. Also between KF_{Nn} and KF_{Wn}, the presence of noise doesn't reduce Kalman filter's performance considerably. The performance can be arranged in the decreasing order as follows: $\text{KF}_{Nn} > \text{KF}_{Wn} > \text{PF}_{Nn} \gg \text{PF}_{Wn}$.

6-3 Translating Terrain

Figure 6-5 shows the simulator setup for a moving terrain to evaluate the prediction performance. The translating mechanism has four links, Base Link being fixed, Link 1 (inside Base Link) translating in z-direction, Link 2 translating in y-direction, and Link 3 translating in x-direction. Similar to the tilting terrain, here also by varying the direction of translation (axes: x, y, xy), magnitude(value: -0.45 to 0.45 m), its rate(value: 0.5-1 m/s), and using various frequencies (range: 0.15-0.6 Hz) of sinusoidal inputs, 16 test cases were generated.

6-3-1 Performance Comparison of Individual Features

From Fig. 6-6a, the performance of individual features can be arranged in the decreasing order as follows: FC, CoG, OPF, IMU, and ZMP. The feature based methods CP, and CPF fall next to CoG, and IMU respectively. As observed in the tilting case, FC and CoG performs better overall, while OPF scores slightly less due to its sluggishness (LFPr - 8/16) in the case of low frequency movements, and IMU performs badly due to its failure (FP - 2/16) to handle high frequency movements. Similar to the flat terrain case, adding a zero

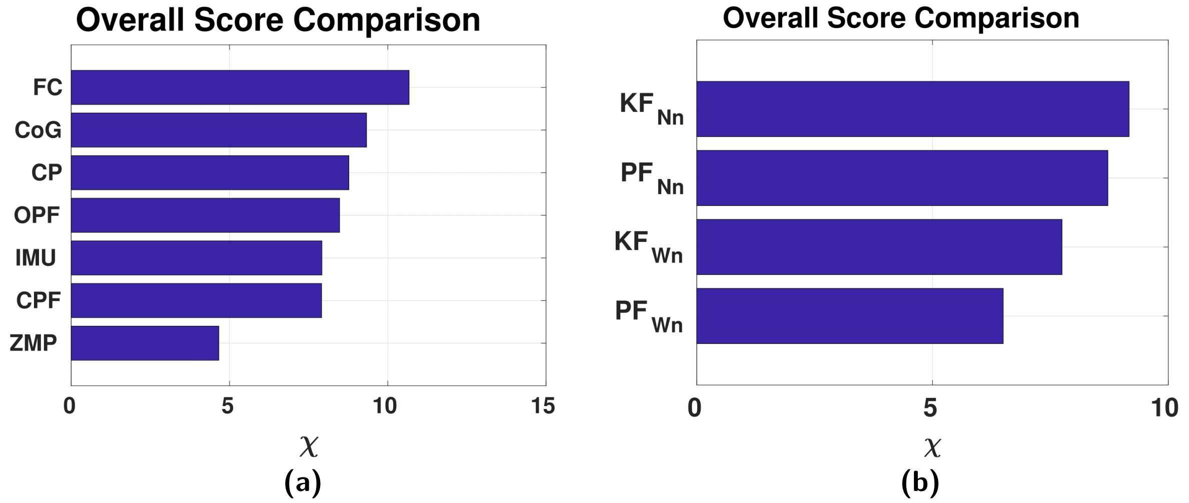


Figure 6-6: Moving terrain prediction performance: (a) single feature-based (without adding noise), and (b) multiple features-based methods (with and without adding noise).

mean white noise (μ=0, σ=0.025) to the sensors related to the best features and repeating the test cases results in the following performance: OPF(5.8) >FC(5) $\gg$ CoG(0.51). This again demonstrates vision's relatively strong performance in the event of noise.

6-3-2 Performance Comparison of Multiple Features

The overall performance of multiple feature-based techniques for translating terrain in the presence and absence of noise, is shown in Fig. 6-6b. The result obtained here are similar to the flat terrain case (Fig. 6-2b). Both in the noise added and noiseless case, Kalman filter performed better than the performance based technique. With external noise added to the system, performance of both KF and PF reduces considerably.

6-4 Tilting Terrain

A tilting platform shown in Fig. 6-7 is used for simulating the tilting terrain. By varying the tilting direction (axes: x, y, xy), magnitude (value: -0.125 to 0.125 rad), its rate (value: -1.5 to 3.0 rad/s) and also using various frequencies (range: 0.1-0.6 Hz) of sinusoidal input, 20 distinct scenarios were generated to test the prediction method.

6-4-1 Performance Comparison of Individual Features

As Fig. 6-8a shows, the best performing features in the case of tilting terrain are CoG, FC, and IMU, which are followed by OPF and ZMP. Better state and contact estimates

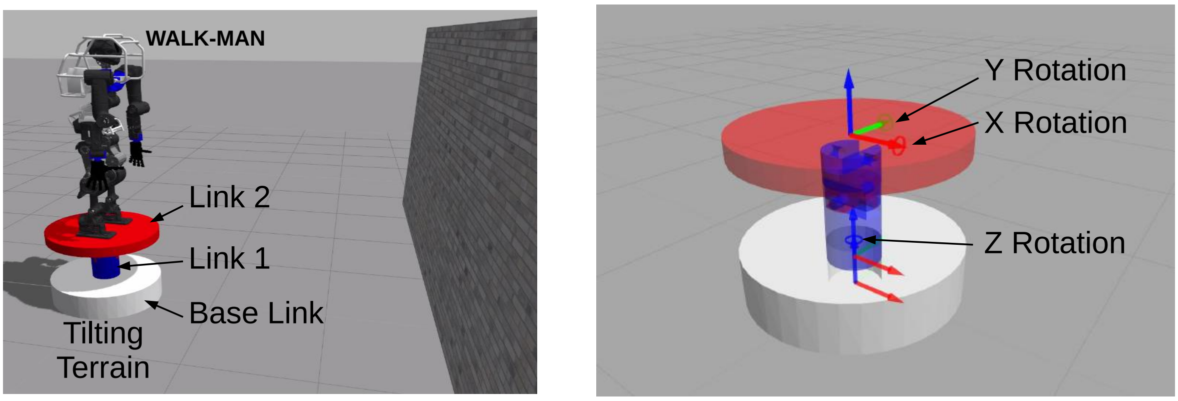


Figure 6-7: Simulator setup for the tilting terrain.

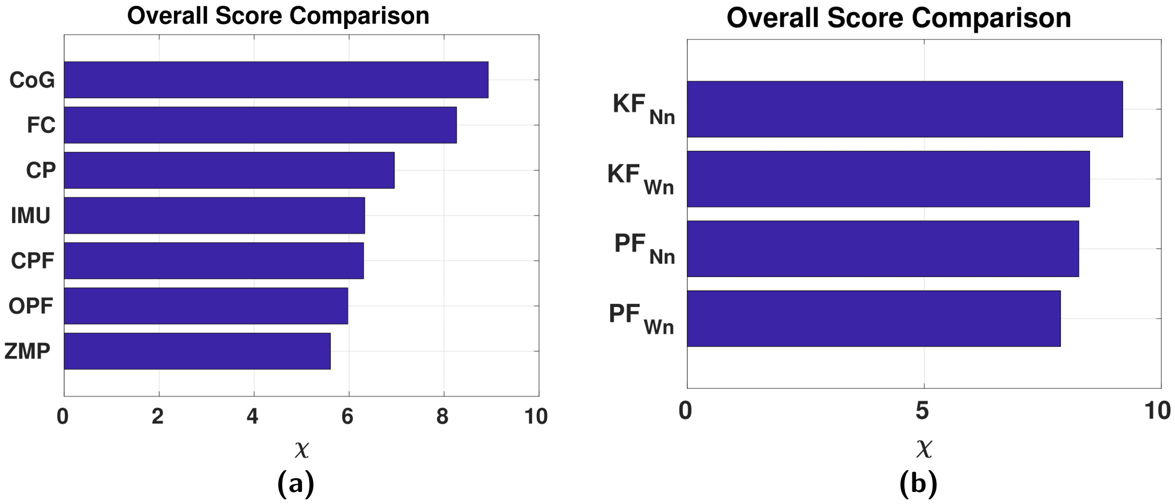


Figure 6-8: Tilting terrain prediction performance: (a) single feature-based (without adding noise), and (b) multiple features-based methods (with and without adding noise).

attributes to the high scores of CoG and FC, while rapid swaying motion accounts for the performance drop in IMU and OPF. As observed in the other cases, CP performance falls next to FC, but its updated version, CPF falls just before OPF.

6-4-2 Performance Comparison of Multiple Features

Figure 6-8b depicts the performance of different multiple feature techniques on a tilting terrain, both in the noise free and noisy case. Similar trend is observed here when compared to the flat and uneven cases. Based on χ, the prediction performance can be arranged in the decreasing order as follows: KF$_{Nn}$ $\gg$ KF$_{Wn}$ >PF$_{Nn}$ >PF$_{Wn}$.

	Single Feature					Multiple Features	
Features / Terrain	CoG	IMU	OPF	FC	CP	PF	KF
Flat	🟡	🟩	🟡			🔴	🟢
UT	🟡	🟩		🔴	🟢	🟩	🟢
TRT	🟩		🔴	🟢	🟡	🟡	🟩
TLT	🟩			🟡	🟡	🟡	🟢

🟢 1st Choice 🟩 2nd Choice 🟡 3rd Choice 🔴 4th Choice

Figure 6-9: Preferred feature and multiple feature-based techniques for different terrains based on the performance metrics.

6-5 Summary of Performance Evaluations

The overall performance of individual features and multiple feature-based techniques over varied terrains are summarized in the table shown in Fig. 6-9. The figure shows the four best choices in predicting the fall over of humanoids in different terrains, based on the performance metrics. We can observe that there is no single feature which can be the primary choice for all four different terrains. The sensor features have varied level of performance on various terrains making it impossible to rely on a single feature to predict the fall reliably in generic situations, and this calls for the multiple feature-based technique. Of the proposed techniques, PF exhibits mediocre results in spite of the technique being based on each individual feature's performance. This is due to the technique's inability to account for the variance of each feature because of constant weights. On the other hand, KF showcases exceptional performance across all terrains by being the primary choice in three out of four different evaluated terrains. KF also performs relatively better in the presence of noise, as shown in the previous sections. This advocates the need to have multiple features and fuse them optimally to obtain a reliable performance across different scenarios.

Chapter 7

Real Robot Experiments and Results

7-1 Experimental Setup

Fall and non-fall experiments were carried out on two different types of terrain using the WALK-MAN along with the other system components integrated with it. The robot was position controlled to stand in a predefined configuration (homing). Different magnitudes of pushes in random direction and at various locations were applied to the robot to evaluate the prediction method's efficacy to differentiate between the fall and non-fall states. To prevent the complete fall over of the robot, its tilting in the x- and y-direction was restricted to around $25°$ using a hoist. Figure 7-1 shows WALK-MAN standing in front of the flat and uneven terrains. As shown in Fig. 7-1b, pebbles are randomly placed to make up an uneven terrain setup.

In order to monitor the evolution of different sensor features and the corresponding Kalman filter-based (KF) ϕ_{KF} while the robot is being disturbed in an arbitrary manner, a custom visualization tool was developed using the components ROS visualization (Rviz) and rqt. While the former is a 3-dimensional (3D) visualizer for displaying sensor data and state information from ROS, the latter is a Qt-based framework used to develop GUI for ROS. The custom visualization tool developed is shown in Fig. 7-2. The top left-side window shows the image features (green dots) automatically selected and tracked to generate optical flow using the ASUS RGB-D sensor. Bottom right-side window displays instantaneous value of center of gravity (CoG) and zero moment point (ZMP) in green and blue spheres, support polygon in red line strips, foot pressure sensor (FPS) cells in white squares, and their corresponding pressure values in blue lines. Finally, the top right-side window shows the evolution of ϕ_{KF} in blue lines against the current time.

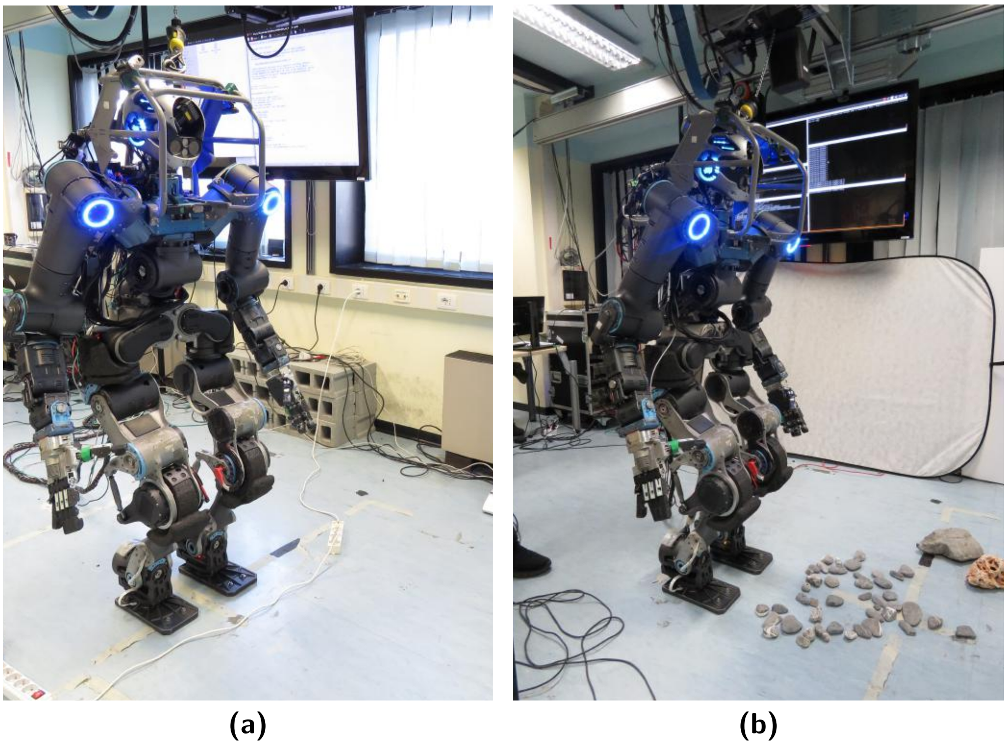


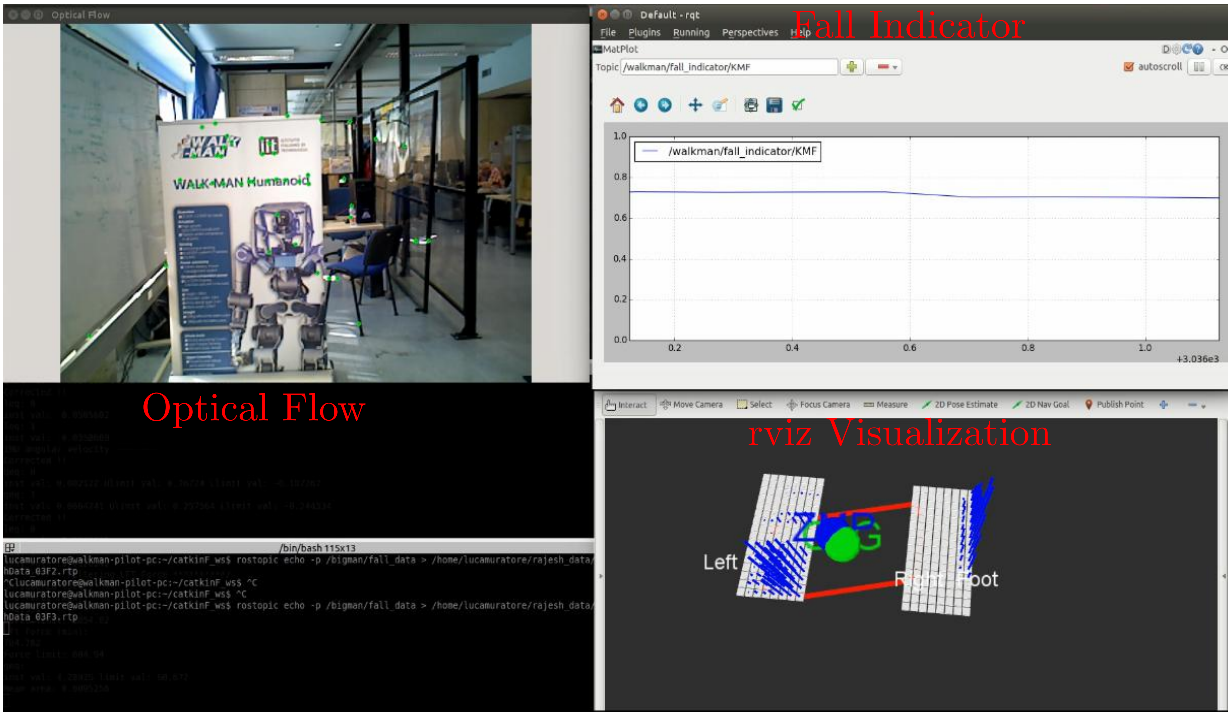
Figure 7-1: WALK-MAN humanoid along with different sensory systems integrated with it is position controlled on two different terrains: (a) flat and (b) uneven terrain.

7-2 Experimental Results

The snapshots taken during the fall experiments carried out with the real WALK-MAN humanoid are shown in Fig. 7-3 and Fig. 7-6. Figure 7-3 shows the robot falling in different directions due to the disturbance applied in the front, back, left, and right side of the robot respectively. Figure 7-6 presents the robot falling in the forward direction for a disturbance applied from behind, while it is standing on pebbles (uneven terrains).

7-2-1 Flat Terrain

Figure 7-4 shows the online evolution of the Kalman filter-based fall indicator ϕ_{KF} and other feature-based indicators, while the robot was being disturbed from different directions. The indicators remains low until significant disturbance is exerted, which is around 50 s, and for subsequent application of major disturbances the indicator value increases proportionately. The sporadic peaks observed during the evolution of single feature-based indicators denote each feature's ability to respond according to the disturbance given to the robot. However, on exerting a much higher disturbance around 80 s, which results in the robot's fall over, the following observations are made: ϕ_{IMU} and ϕ_{OPF} increases significantly but remains well



```
lucamuratore@walkman-pilot-pc:~/catkinF_ws$ rostopic echo -p /bigman/fall_data > /home/lucamuratore/rajesh_data/
hData_03F2.rtp
^Clucamuratore@walkman-pilot-pc:~/catkinF_ws$ ^C
lucamuratore@walkman-pilot-pc:~/catkinF_ws$ ^C
lucamuratore@walkman-pilot-pc:~/catkinF_ws$ rostopic echo -p /bigman/fall_data > /home/lucamuratore/rajesh_data/
hData_03F3.rtp
```

Figure 7-2: Custom visualization tool developed to monitor the evolution of different sensor features and ϕ_{KF} is shown. Top left window shows optical flow, bottom right-side window shows the ZMP(blue sphere), CoG (green sphere), support polygon (red line strip), FPS cells (white cubes), and their corresponding pressure values (blue lines). Top right-side window shows the instantaneous evolution of ϕ_{KF} (blue lines).

below the critical value ($\phi_{\text{critical}} = 0.8$); ϕ_{ZMP} fluctuates around ϕ_{critical} starting from around $60s$, which is probably due to the noisy acceleration values; and the indicators ϕ_{CoG}, ϕ_{FC}, and ϕ_{KF} increase appropriately above 0.8 to predict the fall successfully. During the time interval 50-80 s, ϕ_{ZMP}, ϕ_{FC}, and ϕ_{OPF} remain relatively high when compared to the values of ϕ_{IMU} and ϕ_{CoG}. However, ϕ_{KF} optimally combines the individual indicators based on their noise level and stays little closer to foot contact (FC) since it is the primary feature. Similar to the simulation studies, when $\phi_{\text{KF}} \geq 0.8$ and $F_{mon} > T_m$, the fall prediction module triggers a fall.

Figure 7-5 shows the evolution of ϕ_{KF} for different magnitude of disturbances directed at the robot in arbitrary directions. We can observe that ϕ_{KF} changes according to the disturbance inflicted on the robot and the fall over is predicted once it reaches critical enough to bring the robot down.

7-2-2 Uneven Terrain

For the uneven terrain case, the robot was position controlled to stand on pebbles and the prediction method was started. With the robot being stable, it was subjected to some

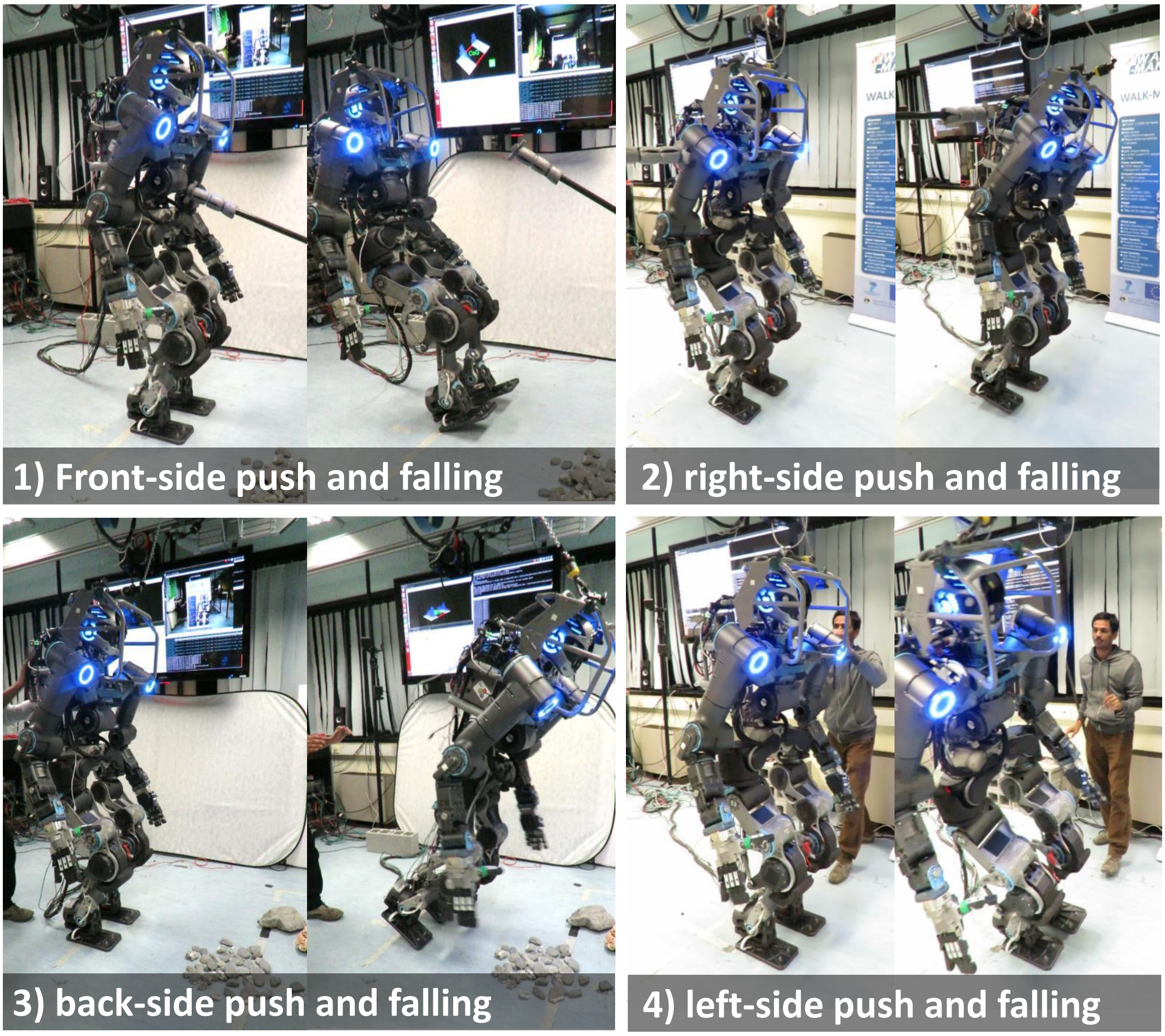

1) Front-side push and falling
2) right-side push and falling
3) back-side push and falling
4) left-side push and falling

Figure 7-3: Snapshots taken during the different direction falls of the humanoid WALK-MAN on a flat terrain.

external disturbances and the method's ability to distinguish them was verified. Figure 7-7 shows the evolution of various ϕ values against time. Since the robot was relatively less stable compared to the flat terrain experiment, the values of ϕ are relatively closer to ϕ_{critical}. During the disturbance period, i.e., 32-58 s, ϕ_{CoG} and ϕ_{IMU} are relatively lower when compared to the values of ϕ_{OPF} and ϕ_{FC}. In the same period, ϕ_{ZMP} gives a false positive (FP) alarm by assuming a value of 1 continuously, which is greater than ϕ_{critical}. This failure can be associated with the high level of noise observed in the acceleration of CoG, which is used in the computation of ZMP. Another observation made during the experiment is the continuous fluctuations of ϕ_{FC} probably due to the noisy unstable contacts, and this makes the prediction of fall over difficult with only FC. Even though

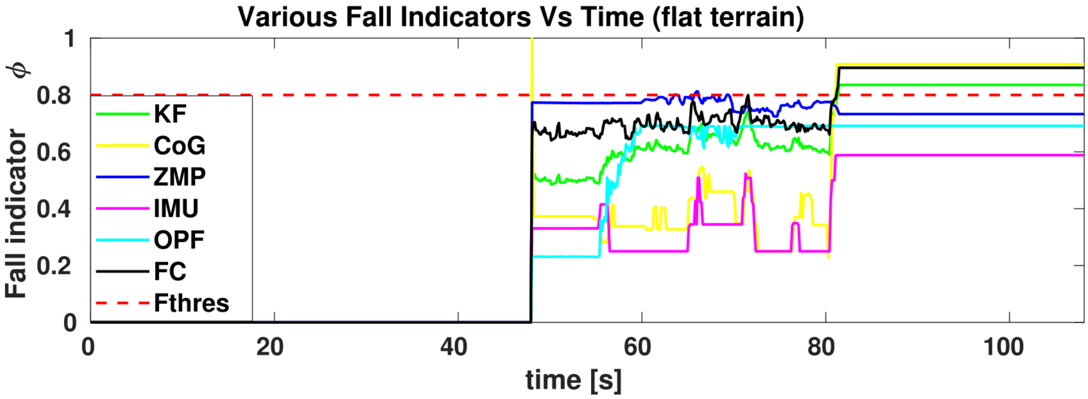


Figure 7-4: Different feature's (CoG, IMU, ZMP, FC, and OPF) fall indicators plotted against time along with Kalman filter-based fall indicator (ϕ_{KF}), for left-side push on the flat terrain.

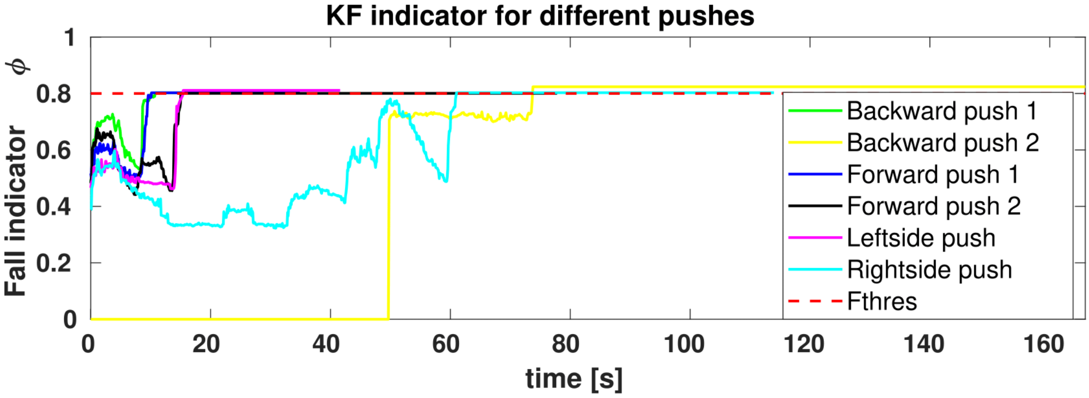


Figure 7-5: Kalman filter based indicator (ϕ_{KF}) plotted against time for different pushing directions on the flat terrain.

ϕ_{KF} uses FC as the primary feature, the relatively stable evolution of CoG and inertial measurement unit (IMU) features lets KF to optimally combine their respective ϕ values. This makes ϕ_{KF} to predict the fall over successfully around 58 s along with ϕ_{CoG} and ϕ_{IMU}.

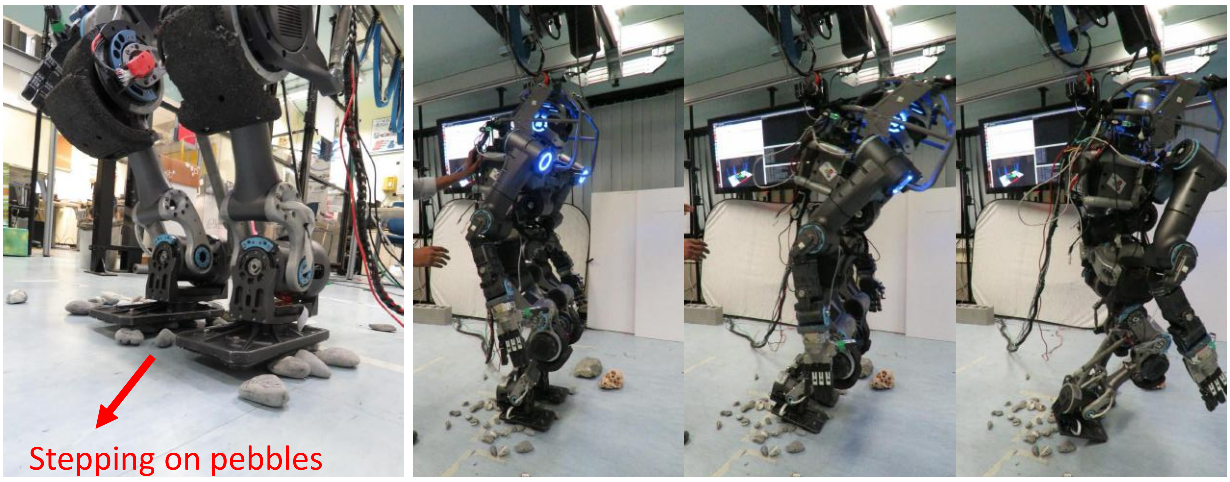


Figure 7-6: Fall prediction experiment carried out on an uneven terrain with WALK-MAN.

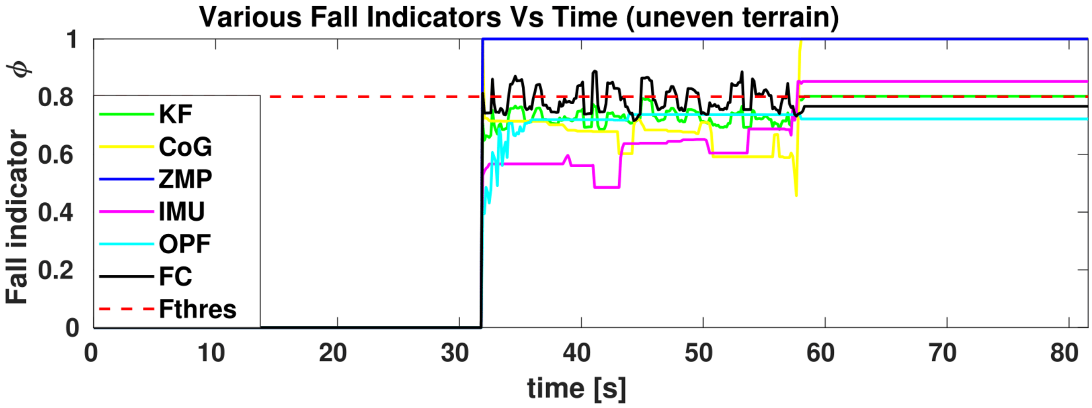


Figure 7-7: Different feature's (*CoG, IMU, ZMP, FC,* and *OPF*) fall indicators plotted against time along with Kalman filter based fall indicator (ϕ_{KF}), for back-side push on the uneven terrain.

Chapter 8

Discussion

Simulation results showed that the performance of individual features varied considerably depending upon the type of terrain, noise-level, etc. Numerical results also showed that Kalman filter-based (KF) technique performs better than individual indicators, conventional instability indicators (capture point (CP), zero moment point (ZMP), etc.,), and performance-based (PF) technique. Even after adding a significant amount of noise to each of the sensors, minimal degradation in the performance of KF was observed. Overall, the results shown in Fig. 6-9 showcases that there are no clear favorites of the five features used. The overall performance can be summarized as follows: center of gravity (CoG) and ZMP are less generic and perform poorly with noise; inertial measurement unit (IMU) is generic but less robust to noise; foot contact (FC) is generic, moderately robust to noise, but involves more time to process data; and optical flow (OPF) is generic, robust to noise, yields high prediction time, but has relatively more false positives. Hence, KF is used to combine the merits and compensate the de-merits of each feature.

For the experiments, minimal tuning of the parameters was required, and it was done by conducting some non-fall experiments on the real system and tuning the parameters by comparing those data to that of the simulator. Overall, the experimental results were satisfactory both on the flat and uneven terrains. In total, 15 experiments were conducted, 13 were found to be successful predictions, and 2 were false positives. On the limitations side, on some occasions loss in the tracking of features was encountered during the swift movement of objects in the camera's field of view. However, this can be resolved with fast and robust tracking algorithms such as FAST[38].

In the determination of prediction window parameters, in particular, the thresholds τ for different feature variables are system and disturbance dependent. This makes it necessary to repeat the data collection experiments to recompute τ. Though the amount of data required to update τ for any arbitrary system had been considerably reduced with the online interpolation technique, automizing it completely can reduce the time consumption even further. Also, considering the online data, i.e., data obtained during the evaluation of the proposed method to update the values of τ could also be advantageous.

In the computation of the fall indicator (ϕ), which determines how far a particular feature variable is from its respective threshold, two types of cost functions were used. Though this reduces the rate of prediction failures such as false positives, false negatives, and late fall prediction, which are prevalent in static threshold based techniques, it is difficult to devise cost functions to capture different nonlinear evolution of various feature variables. However, with machine learning techniques which are good in generalizing any nonlinear function can be used to address this limitation.

For experimental verification of the fall prediction method, only static cases of the robot have been considered. However, this method can be extended to handle fall prediction in robots executing simple whole body manipulation tasks, with minimal parameters tuning. The following section discusses about prediction of fall over during some dynamic movements and presents the results obtained.

8-1 Fall Prediction During Dynamic Motion

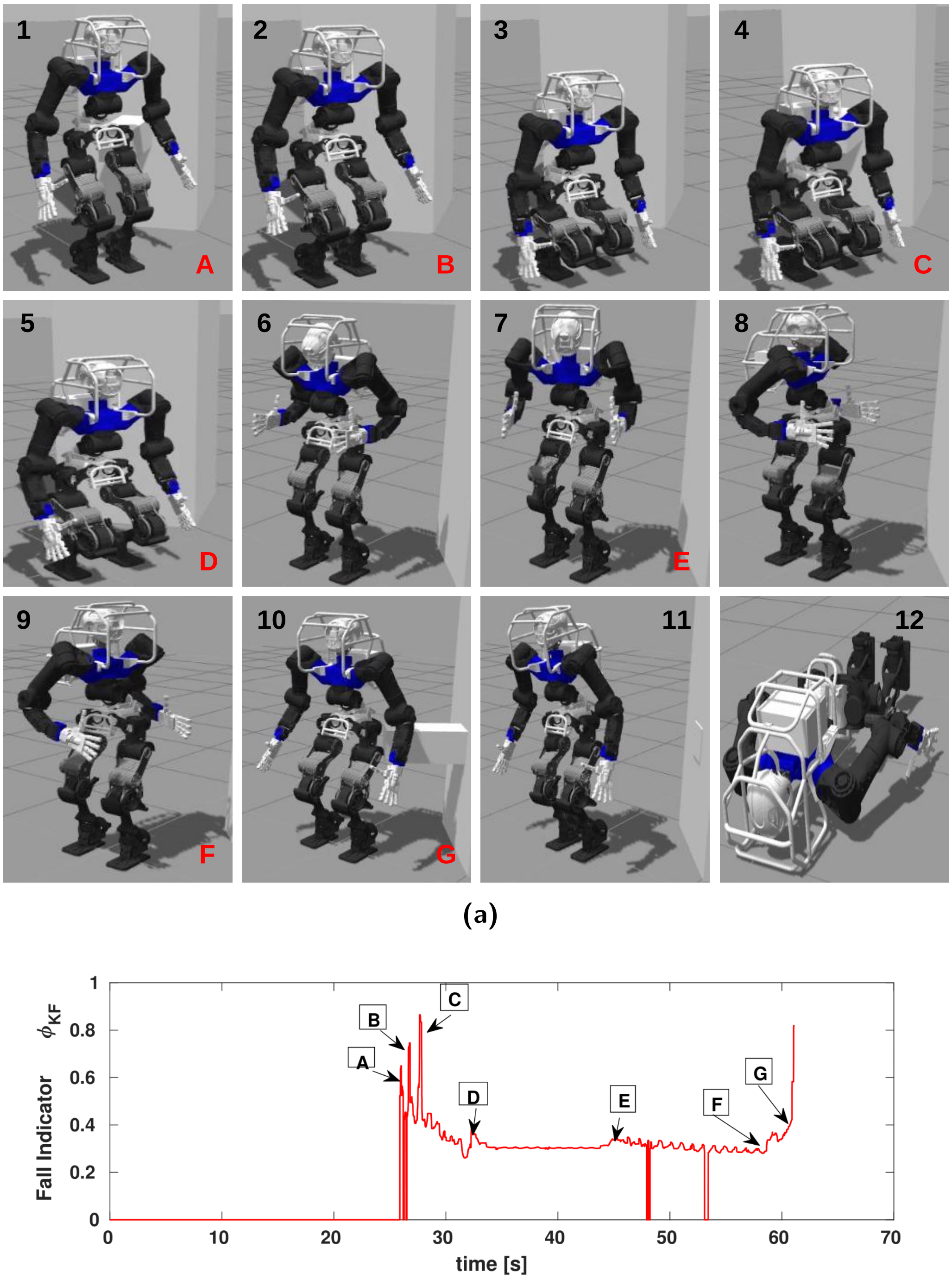
The better performing multiple feature technique, KF is used here to evaluate its fall over prediction performance with a humanoid performing certain dynamic motions.

8-1-1 Dynamic Motions Considered

For the purpose of evaluation, a sequence of whole body dynamic movements were commanded and three different disturbance magnitudes were exerted on the robot, during its motion. Figure 8-1a shows the simulation snapshots taken while the robot executes the commanded dynamic motions. The corresponding evolution of Kalman filter-based fall indicator (ϕ_{KF}) is shown in Fig. 8-1b. The motion sequence include squatting down (B), squatting up with different hand configuration (D), left turn over (E), and back to initial position (F). A sequence of disturbances 400 N (A), 500 N (C), and 1500 N (G) were exerted on the robot for 30 ms at 0.5 m height, as shown in the snapshots 1, 4, and 10 in Fig. 8-1a.

8-1-2 Analysis of Results

In Fig. 8-1b, we can see the KF indicator increasing instantaneously due to the exertion of impulse (A), followed by a slight increase due to the initiation of squatting down motion (B), and this increases further with the impulse exerted on the robot in squat down condition (C). Though $\phi_{\text{KF}} > 0.8$ (fall indicator threshold), it decreases within few seconds, and this instance is perfectly handled by the impulse adaptive counter limit (T_m), as explained



(a)

(b)

Figure 8-1: Prediction performance during dynamic movements: (a) simulation snapshots taken during different motions (B, D, E, and F), and (b) evolution of Kalman filter-based fall indicator ϕ_{KF}.

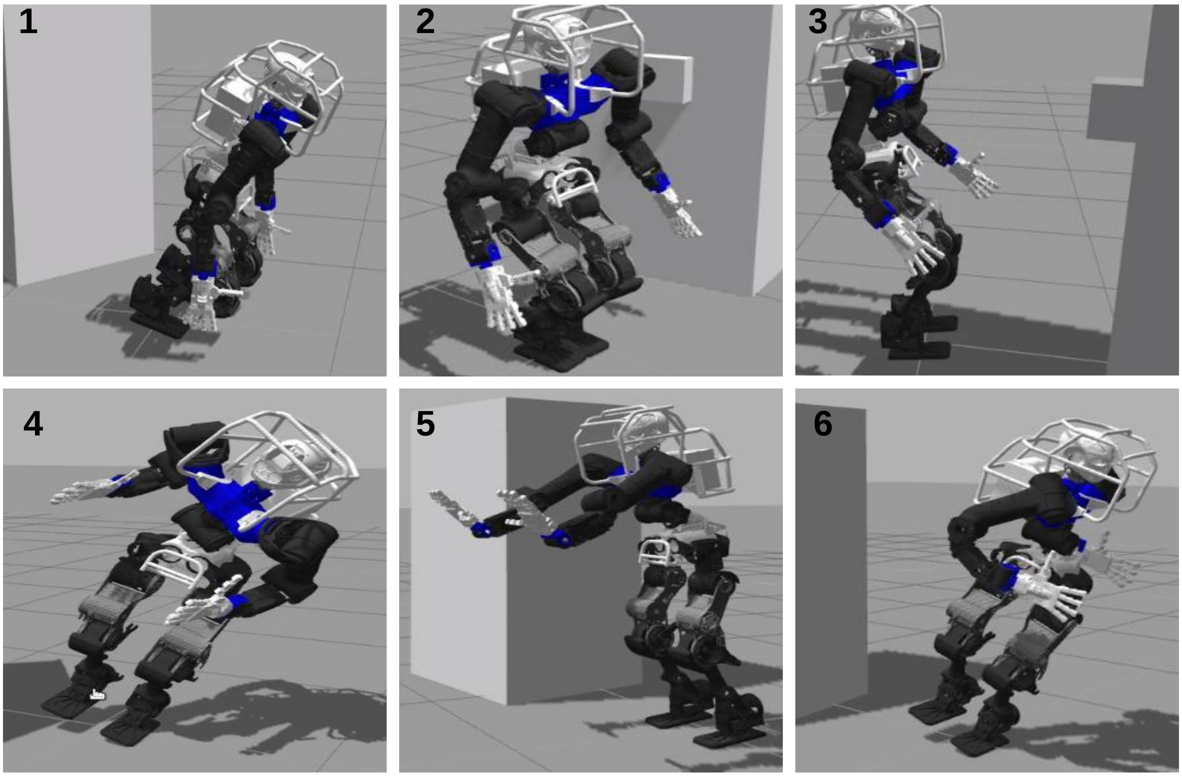


Figure 8-2: Fall prediction with WALK-MAN performing different dynamic movements and varied disturbances applied on it in random directions.

in Section 4-7-1. Following C, minor fluctuation in KF was observed during the different motions carried out at D, E, and F respectively. Finally, at G, a much higher impulse was exerted on the robot while it is getting back to its initial position. This disturbance significantly increases the ϕ_{KF} value to beyond 0.8, and also decreases T_m accordingly to predict the fall successfully. This demonstrates the successful application of the proposed fall prediction method on simple dynamic movements, without tuning any parameters.

Several combinations of dynamic movements involving arbitrary disturbances were simulated to evaluate the prediction performance of KF more extensively. In total, 8 different cases were considered, with 5 FPr and 3 (LFPr) instances. The prediction of fall for some of the cases are shown in Fig. 8-2.

The above results show how the proposed fall prediction method can already be used to predict fall over of humanoids during their dynamic movements. However, for complex movements like walking, regeneration of the thresholds for each feature variable as explained in Section 4-4 need to be carried out.

Part III

Controlled Fall

Chapter 9

Motivation, Related Works, and Challenges

Any desired motion generated by means of active control commands during the fall over of humanoids, which involve either a particular part or the whole body of a humanoid to mitigate the impact forces acting on itself or to avoid inflicting damages to its immediate surroundings is defined as a controlled fall.

9-1 Why Controlled Fall is Necessary?

The objective of controlled fall is to reduce significantly the damages incurred by any humanoid during its fall over. With a reliable, agile, and generic fall prediction developed in the previous part, the next prominent component in fall recovery is the controlled fall. This is important because for the humanoid to recover from its fall over it should be in an operable condition, for which, it has to sustain minimum damages during the fall over, and this is ensured by controlled fall actions developed in this part.

One of the major hurdles for the humanoids to operate in unstructured environments is their fall over. Several fall avoidance strategies have been proposed over the years. A simple feedback control of the tilt angle of the robot's ankle or lowering its center of mass (CoM) was proposed in [17] and this was employed depending upon the disturbance magnitude. In [9], a region called capture region was introduced where the robot has to step to recover from a push to tackle large magnitude of disturbances. Also, an online switching between low-level and high-level push recovery controllers was devised in [39] to generate prevention actions. Nevertheless, falling over of humanoids still remains inevitable, and these falls have a devastating effect on itself and its immediate surroundings, calling for strong measures to deal with it.

9-2 Related Works and Their Limitations

Several works related to the controlled fall have been reported in the past two decades. The previous works can be categorized based on the following techniques:

- Pre-planned Motion Sequence
- Traditional Control
- External Compliance
- Learning Algorithms
- Different Combinatorial Works
- Other Works

In the works reported below, the above techniques are used either individually or in different combinations and accordingly the works are classified and discussed below.

9-2-1 Pre-planned Motion Sequence

Inspired by Ukemi motion, Fujiwara et *al.* started with a pre-planned motion sequence along with external passive compliances, i.e., silicon pads in [40]. This was verified experimentally for backward [18] and forward falls [41] with the former involving some heuristic stages such as squatting, extension, and a touchdown, and the latter with knee bending, breaking speed, and landing. Similar work was also reported in [42], where low-damage fall sequences for humanoid soccer robots were generated based on human motions through iterative manual tunings using custom designed cost functions as a reference. The above works were extended further by proposing an optimal falling motion planner with different reduced models for forward and backward fall overs to reduce the impact force significantly[18, 43]. Similar works combining offline optimization and external passive compliances were also proposed in [44, 45] but the motion primitives were extracted by observing human falling over motions, and they were optimized by making use of an inverted pendulum with a flywheel. In the aforementioned works, the optimal fall over motions were generated offline with direction-specific reduced models, making it infeasible to adapt for any changes in fall directions, disturbance magnitude, and the robot's state.

9-2-2 Traditional Control

Few works relying only on control techniques were proposed and verified successfully with real systems in few cases. Some of them include a backward falling technique [16] generated using an inertia shaping control to land on the robot's backpack which could withstand considerable impacts during the fall overs. Several optimal control technique based works such as a whole body trajectory optimization using Legendre-Gauss-Lobatto (LGL) method to reduce the impact forces by means of cost functions [46], direct shooting method to determine the optimal hand location to stabilize the robot's motion and minimize the impact forces [47], and a 3D symmetrization method to generate optimal falling over motion online [48] were proposed and verified to reduce impact forces significantly. Though the generated motions were online and optimal, they were system specific[16], limited to certain disturbance direction range[16, 46], and lacked the adaptability to different impulses[46, 47, 48].

9-2-3 External Active Compliances

In addition, a unique impact reduction concept based on actively controlled external compliance was proposed in [49]. This included an airbag system to reduce the impact acceleration of falling over humanoids. Two dummy models were designed and fabricated to verify the extent of damage reduction with conventional shock absorbing materials and the proposed airbag system. The latter's performance was found to be better with the acceleration reduced from 80G to 22-25G, reducing the impact force considerably. Minor damages on the on-board system and neck joint of the robot were reported. Similar external active compliance in the form of inflating pads was also reported in [50]. Here the pads were attached to the robot's palm and the arms were position controlled to make contact with it during the robot's fall over. Unlike the previous method, the proposed one was verified with a partial fall over of a humanoid.

9-2-4 Learning Algorithms

Overall, the aforementioned literature reported either planned motion or carefully designed controllers comprising different strategies. In an attempt to automate control actions, few learning based techniques emerged such as Markov decision process based multiple contact planning [51], wherein a series of abstract models are used to plan the location of different contacts and their specific sequence to reduce the impact. The contact sequence in particular is planned using Markov Decision Process (MDP) which minimizes the total cost associated with impulse. Finally a feasible whole body motion is generated by formulating an optimization problem which tries to match the existing contact, CoM position and the next contact to their respective optimal values. In another work [52], an actor-critic neural network architecture is proposed to simultaneously select the best body part for making contact, its location, time and the joint torque to be applied. Discrete problem of choosing the best contact and continuous problem of joint torque optimization are done simultaneously. In the above works only sagittal falls are addressed, in particular, forward fall; and most of the methods (except for [52]) require training over a significant number of trials, limiting their chances to apply online and generalize them to handle different direction falls.

9-2-5 Different Combinatorial Works

Several other interesting combinatorial works were proposed to mitigate the impact forces during the fall over of humanoids. In [53], a planned motion sequence involving crouching of legs, elongation of arms, and touching down of knees was combined with external passive compliance in the form of elastic energy absorbing mechanical structure and with the internal compliance control of arms. Very similar to the previous work but the two types of compliances were combined with a posture reshaping technique in [54]. Another interesting combination is the usage of learning and traditional control algorithms as in [55], where the conversion of potential energy to kinetic energy is limited by generating whole body motion to make a tripod contact with two hands and a leg. The optimal contact locations were determined through reinforcement learning. Interestingly, the above work was combined parametrically with a human-inspired active internal compliance, and the parameters were optimized together through policy gradient reinforcement learning to reduce and absorb the impact forces [56]. Finally, there is also a work combining the pre-planned motion sequence of hands with the fuzzy logic control of arm compliance [57]. The former is to prevent ground contact of the vital parts, and the latter is to absorb the impact forces. Most of the above combinatorial works address only sagittal falls and are also prone to other drawbacks such as direction specificity, lack of generality to handle different impulses, offline training requirement, no quantitative comparison between similar methods, and heuristic-based control.

9-2-6 Other Works

Apart from the above works, there are also some different kinds of controlled fall works reported to prevent damages to the immediate surroundings of a humanoid, when the robot falls over. In reference to this, a fall direction changing controller was proposed in [58] for humanoids by means of intelligent stepping and inertia shaping techniques. The former technique changes the position and orientation of the edge along which the humanoid falls over, and the latter modifies the robot's centroidal inertia and thereby indirectly changes its falling over direction. The proposed technique was later extended to handle the complete 360° falling directions of humanoids in [59].

9-3 Motivations and Challenges

The main motivation to control the fall over of humanoids is to reduce the damage it can sustain or bring it upon their surrounding objects as discussed in Section 1-2. Further, humanoid robots being expensive, in particular, human-sized ones, severe damages can incur high repair cost and also keep them away from achieving their design purpose, i.e., to operate autonomously in real environments. Knowing the significance of addressing this issue several works had been done so far as summarized in the previous section. But still the fall over issue hasn't been addressed completely due to the following limitations which are prevalent in the existing works:

- Omnidirectional fall strategy: Most of the present controllers mainly deal with sagittal falls, and very few have addressed frontal falls. Even experimental verification of the proposed controllers has been reported only for sagittal falls in the case of human-sized humanoids. In reality, the fall over of humanoids can be in any arbitrary direction depending upon the nature of disturbances. To handle such situations, there is an imminent requirement of the controllers to sense the directions of fall over and act accordingly.
- Disturbance generic controllers: Numerous simplified models are used in the contemporary controllers making them immensely difficult to generalize them to handle different direction fall overs in real time.
- Online planning and adaptation: Planning and adaptation of the controllers to different disturbance impulses, robot's state, etc., is another vital trait of the controller, for it to be agile and efficient in reducing the damages. Though some of the controllers are successful in planning the controlled fall motions online, they lack the ability to adapt to instantaneous situations.
- Motions closer to "break-fall": Several existing break-fall motions such as the Ukemi, parkour roll, paratrooper landing fall, etc., are proven techniques to reduce the impact forces during the fall over of humans. Though there have been few controllers proposed for humanoids inspired by the aforementioned techniques, only a few are capable of generating dynamic motions, and none of them are still close to those of break-fall motions.

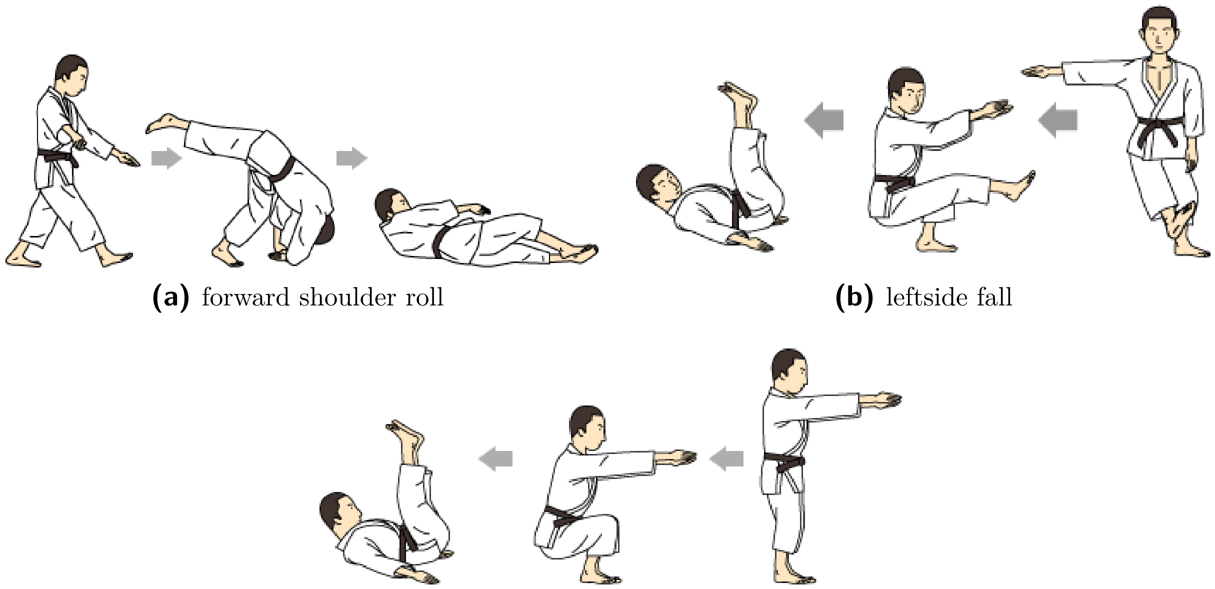
The above limitations are the main motivations for us to continue to work on the controlled fall over of humanoids.

9-3-1 "Break-fall" motions

There are several break-fall motions prevalent among trained professionals in different fields. These techniques have been proven among humans to successfully reduce the impacts they sustain during their fall over. Since the novel controllers proposed in this work to handle humanoid fall overs are mainly inspired by some of those techniques, a brief description about few of them are given below:

Ukemi

One of the most notorious break-fall technique is Ukemi, which translates to 'receiving body or self'. It refers to a set of motion techniques practiced by martial arts people, in



(a) forward shoulder roll

(b) leftside fall

(c) backward fall

Figure 9-1: Ukemi motion techniques practiced by judo players for different direction fall overs to reduce the damages sustained.

particular, judo players, during their fall over in different directions to protect their body from any major damages. The ground contact is always made with body parts which have a high surface area of contact such as shoulder back, hip, and butt. Some of the techniques adopted during the fall over in different directions are shown in Fig. 9-1.

Parkour Roll

Another break-fall technique is parkour motion which originated out of military drills and is prevalent now among street performers, skateboarders, etc. Figure 9-2 shows one of the parkour roll motions executed after jumping over a rail. Though parkour refers to a set of obstacle avoiding motions, parkour roll deals with the fall over which is common during or after the obstacle avoiding maneuvers. The rolling is very similar to the diagonal shoulder roll observed with Ukemi, as a result, it is also effective in reducing the impact forces during the fall over of humans. However, unlike Ukemi, the rolling motion is adapted according to the direction of fall over.

Paratrooper Landing Fall

It is a break-fall technique mostly practiced by airborne soldiers called paratroopers in the military. During the paratroopers landing, they descend down at an average velocity of 8m/s, significant enough to cause damage to their legs. This technique is adopted by the

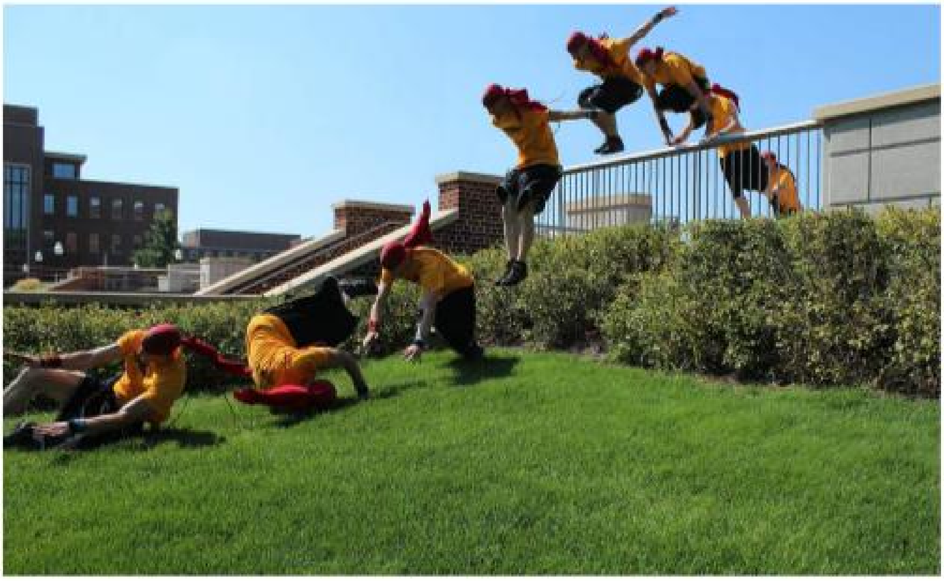


Figure 9-2: Parkour rolling motion executed to minimize the damage due to falling over after overcoming an obstacle.

paratroopers at their landing phase to land smoothly on the ground. It involves restraining the knees and ankles together, touch the ground with flat feet, followed by pushing and turning the body to roll over the back as shown in Fig. 9-3.

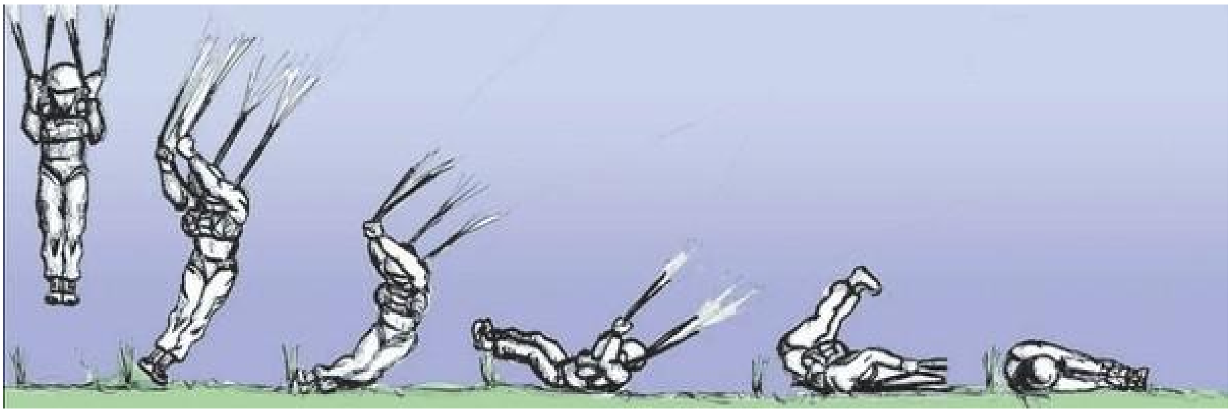


Figure 9-3: The fall over motion technique adopted by the paratroopers during the landing.

9-3-2 Challenges to Face

In addition to the aforementioned motivations, there are also a few challenges which need to be given proper consideration in addressing the ongoing issue. The two major issues are the following:

- Critical evaluation of present controlled fall techniques: There have been numerous works reported involving different combinations of various techniques as mentioned in Section 9-2. Some of these works have been applied to real systems and a considerable reduction in impact forces during fall over has been reported. However, there hasn't been any active comparison or critical evaluation of the proposed controllers, doing so can help us in determining which technique or combination of techniques significantly reduces the impact forces.

- Quantification of damage: The extent of damage for the same amount of force depends significantly on the material properties of the part on which the force is exerted. Hence, reduction in impact force doesn't necessarily guarantee if the damage sustained due that force is less enough for the robot to continue to operate. This necessitates a method to quantify the damages sustained by humanoids in order to evaluate the performance of any proposed controlled fall techniques efficiently.

9-4 Contributions

In this part, we propose two novel controllers using the energy concepts inspired by UKEMI motion. The energy of falling over systems, its relation to the impact force, and the energy concepts are discussed in detail in the sections to follow. The energy concepts considered here for the controllers are as follows:

- Energy minimization - minimizing the total energy of the system.
- Energy distribution - distributing the minimized energy over multiple contacts.
- Energy injection - injecting energy into the system to ensure consecutive contacts.

The first controller as shown in Fig. 9-4 is a combination of the energy concepts: minimization and distribution. Unlike [55], where energy conversion has been used to backup control measures and later for evaluation, we have actively employed the energy in designing our control law and also used it later for critical evaluation. The main contributions made here are as follows: first, an energy shaping based control law for falling motion generation is developed to minimize energy (ES); second, energy distribution polygon is proposed to distribute the minimized energy over multiple contact points by reconfiguring the end-effector pose (ED); and finally, an online whole body control exploiting energy shaping and distribution notions is proposed with a single reduced model to make affordable sagittal and lateral falls. Additionally, an orientation control of the arms is employed to reduce possible damages to the hands, which is often not addressed in previous works.

Though the motions generated with ES + ED type controller reduced the impact forces considerably, the motions were relatively static when compared to Ukemi. To generate a more dynamic motion a new type of controller was devised by combining all three energy

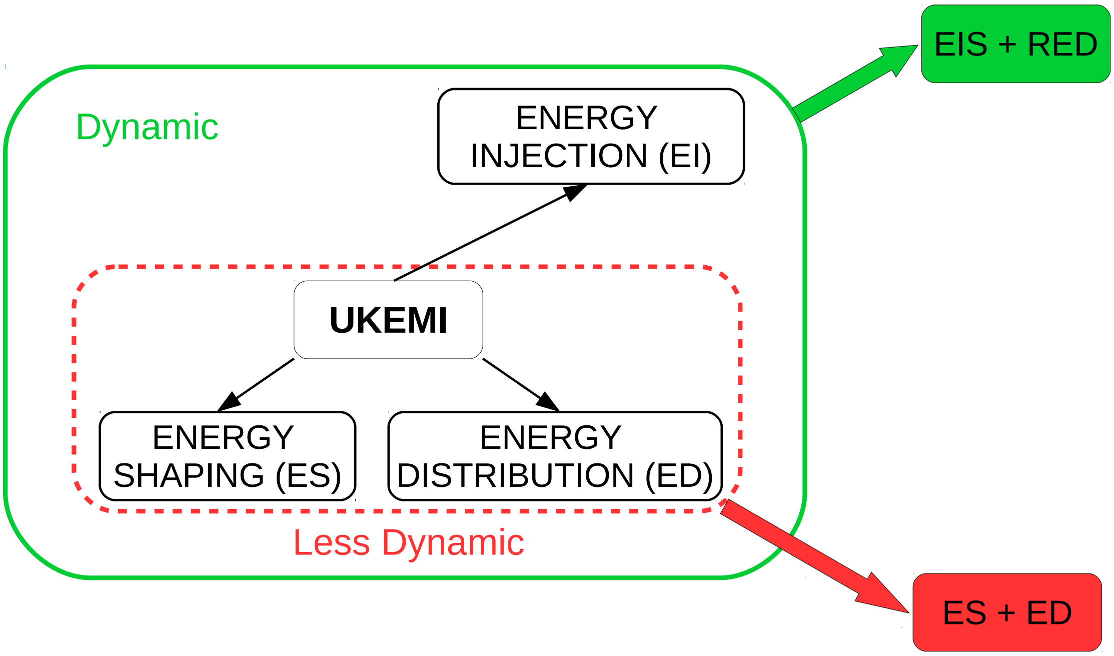


Figure 9-4: The two novel controllers, ES+ED and EIS+RED, proposed to reduce the impact forces during the fall over of humanoids using the energy concepts inspired by UKEMI. ES+ED combines energy shaping and energy distribution techniques, while EIS+RED includes energy injection apart from the aforementioned ones.

concepts resulting in EIS+RED type controller. There have been some previous works which are inspired by Ukemi [40, 18, 41], and some capable of generating dynamic motions during fall over [51]. However, the proposed method is different due to its following novel traits: determination of critical rolling parameter (CRP) for optimal rolling configuration based on a rolling study, online rolling controller (ORC) to compute the suboptimal values for CRP and to regulate the system's energy, energy-based control actions, and notion of energy distribution polyhedron (EDPH) to realize the sub-optimal rolling posture. In the following sections, the aforementioned controllers will be explained in detail, and their performances will be analyzed critically.

Chapter 10

Energy in Falling Over Systems

Energy is a quantitative property which a system must either possess or transferred to it in order to do any work. Since energy is a conservative quantity, it can neither be created nor be destroyed but can only be transformed from one form to another. The two main forms of energy are potential and kinetic, although there are several other forms such as thermal, electric, magnetic, nuclear, etc. While the potential energy (PE) of a system represents the amount of work which can be done by its weight, the kinetic energy (KE) denotes the same, but it is associated with the system's speed. Both the energies combined together form the total mechanical energy of the system and this can be manipulated in the desired way to reduce the impact forces during the fall over of any systems. The following sections explain briefly how energy and impact force is related, the evolution of energy in falling and rolling systems, and how it can be controlled in a desired way.

10-1 Relation Between Energy, Impact Force, and Deformation

The basic relationship between the $\mathrm{F_{imp}}$ and the energy dissipated (ΔE) can be derived by considering the collision between body B and ground G shown in Fig. 10-1. Assuming the initial energy of B to be $E_i = mgh + \frac{1}{2}mv^2$ and it undergoes a plastic collision with G resulting in the end energy $E_e = 0$. With a deformation of δd observed in G during the collision, $\mathrm{F_{imp}}$ and ΔE can be related as

$$\mathrm{F_{imp}} = \Delta\mathrm{E}/\delta d, \tag{10-1}$$

where $\Delta\mathrm{E} = E_i - E_e$ represents the energy dissipated instantaneously during the ground contact .

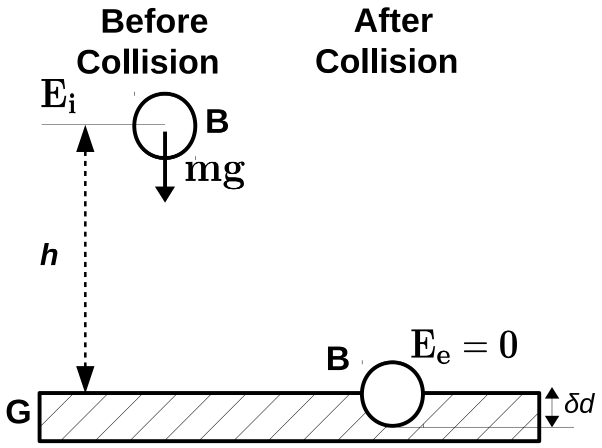


Figure 10-1: Collision between body B and G due to the free fall of B.

From the above equation, it can be noticed that there are two ways to reduce F_{imp}. Since s is inversely proportional to F_{imp}, by increasing δd it is possible to reduce F_{imp}, and this can be done by either adding external compliances such as silicon pads, airbags, etc. or by incorporating internal active compliances in the form of compliant joints, arms, etc. The second method is to minimize the instantaneous dissipation of energy (ΔE) to reduce F_{imp}. This can be done by decreasing E_i, energy just before the collision, or distributing ΔE over multiple contacts in order to reduce ΔE/contact which in turn reduces F_{imp}. This method has been exploited heavily in the controllers which are to be discussed in the chapters to follow.

10-2 Insights on Rolling

In the previous section, we saw that instantaneous dissipation of energy can be reduced by either minimizing the initial energy of the system or distributing it over multiple contacts. Making such contacts in a consecutive manner and smooth transition between each one of them results in a rolling motion. This section, in particular, gives some insights on how a system's energy influences its rolling and how it evolves during such motion.

For simplicity, a planar example is considered for illustration as shown in Fig. 10-2. It includes two planar point mass (m) systems, a circle (S_1) and a heptagon (S_2). Here, S_2 is used to represent the rolling motion of a humanoid in 2-dimensional (2D). Let $\mathbf{u}_i$, $\mathbf{f}$, ω, $g = 9.8$ m/s^2, and e_{ke}^{i-} represent initial velocity, force acting on the system, angular velocity of rolling/tipping, acceleration due to gravity, and initial kinetic energy of the system respectively. Assuming energy loss due to friction is less, S_1 with e_{ke}^{i-} can make continuous rolling contacts such as C, E, ... with minimal loss of energy.

However, for S_2 with e_{ke}^{i-}, it can only tip over E, if it satisfies the following *energy rolling condition*:

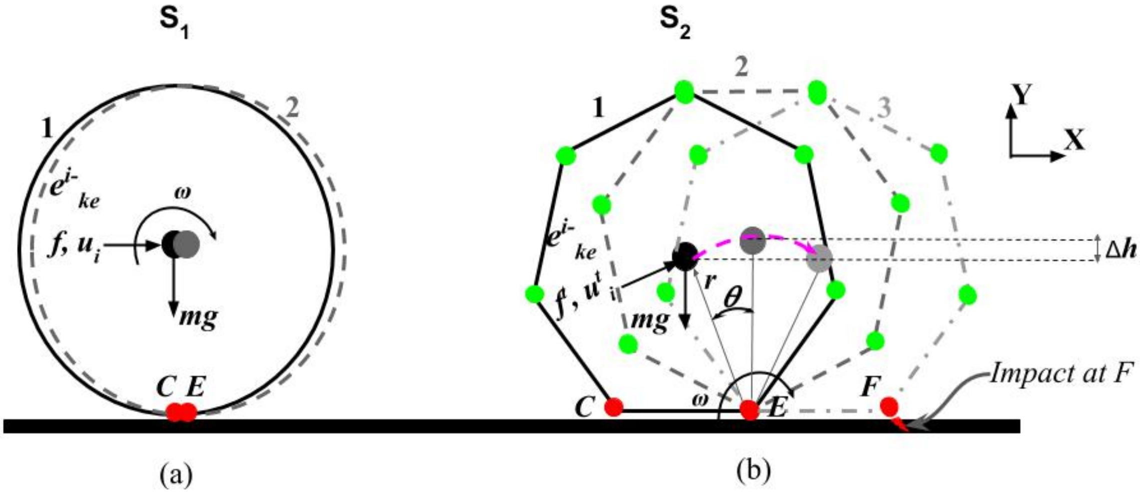


Figure 10-2: Dynamics involved in the rolling of two systems: (a) circle and (b) polygon (heptagon).

$$e^{i-}_{\mathrm{ke}} \geq w_{\mathrm{E}}, \tag{10-2}$$

where $w_{\mathrm{E}} = \mathrm{f}^{\mathrm{t}} \cdot r\theta$ is the work to be done by S_2 against g to move for an arc length of $r\theta$ to reach the state 2, and f^{t} is the tangential force to the arc. In situations where it is difficult to determine the force acting on the system (f), w_{E} can be calculated as the change in potential energy, i.e., $mg\Delta h$, where Δh is the difference in CoG's height between state 2 and 1.

For consecutive tip overs to be successful, the energy rolling condition (10-2) has to be satisfied. Further, because the consecutive rolling contacts made with S_2 is more discontinuous when compared to S_1, the energy is lost in the form of heat or deformation due to the impact at each contact. For instance, Fig. 10-2b shows the tipping over motion from the state 2 to 3, resulting in an impact at F. Due to this impact, the energy of S_2 after impact at state 2 can be computed as $e^{2+}_{\mathrm{ke}} = \epsilon\, e^{2-}_{\mathrm{ke}}$, where ϵ is the coefficient of restitution defining the amount of energy lost during the impact between two bodies. This reduction in energy can lead to the violation of energy rolling condition, and this may halt the system abruptly. Interestingly, the above condition can be avoided by injecting additional energy into the system, more of which is discussed in Chapter 12.

Chapter 11

Falling Over Controller

In this chapter, of the two controllers proposed in Section 9-4 to mitigate the impact forces sustained by humanoids during their fall over, the first controller (ES + ED) is presented here in detail. The rest of the chapter is organized as follows: Section 11-1 talks about the principle behind the proposed controller. The derivation of energy shaping control law and its numerical evaluation is discussed in Section 11-2, followed by the description of an algorithm to construct energy distribution polygon for different direction fall overs in Section 11-3. The above two techniques are combined to form a whole body control scheme in Section 11-4, and the simulations results and comparisons are carried out in the Sections 11-5, 11-6, and 11-7.

11-1 Falling Over Principle

Considering the limitations mentioned in Section 9-2, in this chapter, we strive to propose a controller which has the following traits: instantaneous control action generation, online applicability, and versatility to handle different direction falls. To achieve this, we draw our inspiration from humans trained to break falls, e.g., Judo player's Ukemi motion. This motion has been shown to be successful in damage reductions during the fall overs of humans and widely recognized as the benchmark, ever since the inception of controlled fall concept for humanoids [40]. The main reason for its wide success is mainly related to energy, and specifically, we believe that the following principles play a vital role:

- energy *minimization*—squatting/crouching motion observed in the beginning of Ukemi and verified to be effective in [37];
- energy *distribution*—rolling motion of Ukemi to distribute energy over multiple contacts right after the crouching pose [60]; and

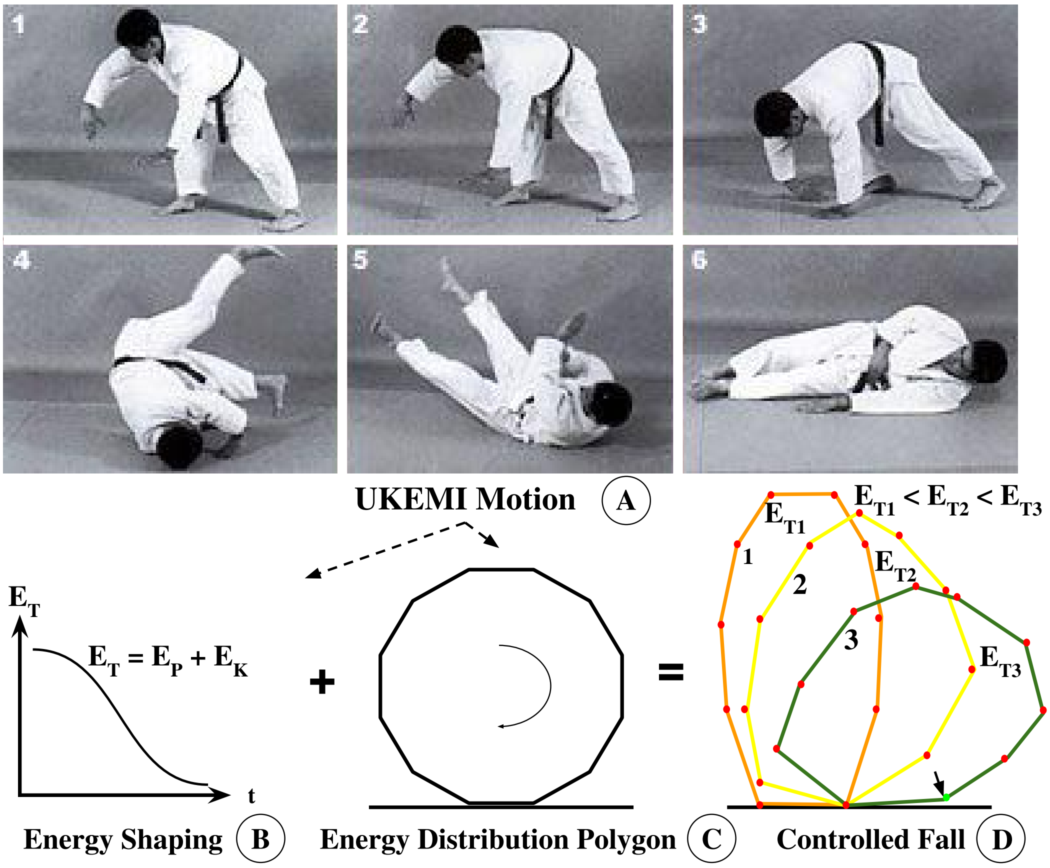


Figure 11-1: Concept of the proposed energy-based falling over control, inspired from the Ukemi motion (A): energy shaping (B) and energy distribution polygon (C) result in controlled fall motion (D), where E_P, E_K and E_T represent the potential, kinetic and its total energies, respectively, and E_{Ti} denotes the total energy of polygon i.

- energy *absorption*—muscle contraction and soft tissues [61] and joint compliances.

In this work, the first two strategies are explored in detail. First, with the minimization principle, a non-linear control technique called energy shaping is used to generate energy based control law to reduce the total energy of the system. Though energy shaping had been introduced in [62] and widely used for under-actuated systems control [63, 64], usage of energy shaping for controlled fall of humanoids has not been studied. Second, an energy distribution polygon is proposed to distribute the minimized energy over multiple contact points, and finally, the two principles are combined together to generate the desired controlled fall motion which mitigates the impact forces sustained during the fall over of humanoids as shown in Fig. 11-1.

11-2 Energy Shaping Method

The basic principle here is to minimize the energy using the non-linear control technique energy shaping as shown in Fig. 11-1. Assuming a constant deformation (δd), minimizing the system's energy is hypothesized to reduce the $\mathrm{F_{imp}}$ based on the relationship between $\mathrm{F_{imp}}$ and the dissipation of energy (ΔE) shown in Section 10-1. This section synthesizes an energy shaping control law by monitoring the current and the desired energy level of the system. The proposed control law is then evaluated with the telescopic inverted pendulum (TIP) model.

11-2-1 Proposed Energy Shaping Control Law

Humanoids in their stable configuration possess most of their initial energy in the form of potential energy (E_P), and during fall this gets converted to kinetic energy (E_K) resulting in $\mathrm{F_{imp}}$. As explained earlier, $\mathrm{F_{imp}}$ can be reduced by decreasing E_i, which in humanoids involves mainly the regulation of its center of gravity (CoG) and this is achieved here using the nonlinear control technique, energy shaping. The controller proposed here, in particular, is inspired by the swing-up control for simple pendulums[65]. The simplest model considered here for devising the control law and its evaluation is TIP shown in Fig. 2-7b, since it can represent both the humanoid's free fall and its lowering of CoG. In Fig. 2-7b, m and r denote the mass and variable link length of the pendulum respectively, while g represents the acceleration due to gravity. With respect to the humanoid, m represents its total mass (assuming point mass) while r denotes the distance between its CoG and the tipping point.

The equation of motion of the telescopic inverted pendulum can be written as given below:

$$m\ddot{r} + mg\cos\theta - mr\dot{\theta}^2 = 0, \tag{11-1}$$

$$mr^2\ddot{\theta} + 2mr\dot{r}\dot{\theta} - mgr\sin\theta = 0. \tag{11-2}$$

Assuming the pendulum is free falling due to g, and r is the only variable that can be controlled, the optimal input to minimize the dissipation of energy and thereby reduce the impact force can be derived as follows:

The total energy of any system E is given as, $E = E_K + E_P$. The initial total energy of the TIP model can be written as

$$E_i = \frac{1}{2}mr^2\dot{\theta}^2 + \frac{1}{2}m\dot{r}^2 - mgr\cos\theta. \tag{11-3}$$

Taking the desired energy (E_j) just before the impact to be 0, the difference in energy $\tilde{E} = E_i - E_j \implies \tilde{E} = E_i$. Differentiating $\tilde{E}$ with respect to time we obtain $\dot{\tilde{E}}$ as

$$\dot{\tilde{E}} = \frac{1}{2}m[(2r^2\dot{\theta}\ddot{\theta}) + (2r\dot{r}\dot{\theta}^2)] + \frac{1}{2}m(2\dot{r}\ddot{r}) - mg[(-r\sin\theta\dot{\theta}) + \cos\theta\dot{r}]. \tag{11-4}$$

Substituting $\ddot{\theta}$ from (11-2) in (11-4) and neglecting the higher order terms in r (i.e., $\ddot{r} = 0$), $\dot{\tilde{E}}$ can be written as

$$\dot{\tilde{E}} = 2mgr\dot{\theta}\sin\theta - (mr\dot{\theta}^2 + mg\cos\theta)\dot{r}. \tag{11-5}$$

Define $\dot{r}_d$ as the desired link velocity of the pendulum since $\dot{r}$ is the only controllable variable in (11-5) and also taking $A \triangleq 2mr\dot{\theta}g\sin\theta$ and $B \triangleq mr\dot{\theta}^2 + mg\cos\theta$, we get the following:

$$\dot{\tilde{E}} = A - B\dot{r}_d. \tag{11-6}$$

To maintain $\dot{\tilde{E}}$ negative and make it dependent on $\tilde{E}$, we propose the following control law

$$\dot{r}_d = (kB\tilde{E} - A/B), \tag{11-7}$$

where $k > 0$ is a control gain. Applying control (11-7) to the system in (11-6), we obtain the closed loop dynamics as follows:

$$\dot{\tilde{E}} = -kB^2\tilde{E}. \tag{11-8}$$

From (11-8) it is evident that with the chosen input $\dot{r}_d$, the gradient of energy difference will always be decreasing resulting in lower $\tilde{E}$, which in turn guarantees the reduction of $\mathrm{F}_{\mathrm{imp}}$. The computed $\dot{r}_d$ is used to control r of the TIP model, while for humanoids it is used to control its CoG.

11-2-2 Numerical Evaluation With TIP

With $m = 1$ kg, $r = 1.15$ m, $k = 0.005$, and initial $\theta = 0.025$ rad, the TIP model is simulated with control law (11-7). Figure 11-2a and 11-2b shows the energy plots obtained for TIP model in the case of free fall and controlled fall with energy shaping. With free fall, as the pendulum hits the floor complete E_P is converted to E_K and hence $\Delta\mathrm{E} = 11.3$ J. In the case of controlled fall, the energy in the beginning is mainly associated with E_P, hence $\dot{r}_d$ is negative (i.e., retraction) until $t = 1.2$ s to bring down E_P. After that, E_K starts increasing mainly due to $\dot{\theta}$, which results in positive value of $\dot{r}_d$ (i.e., extension) to reduce $\dot{\theta}$. With controlled fall $\Delta\mathrm{E} = 3.147$ J, a reduction of 72% is observed.

The CoG motion of the TIP model with the proposed fall control law is shown in Fig. 11-2, where both the retraction and extension of TIP are clearly visible. Interestingly, the online controlled CoG movement is similar to the fall motion generated based on an offline optimization method in [66].

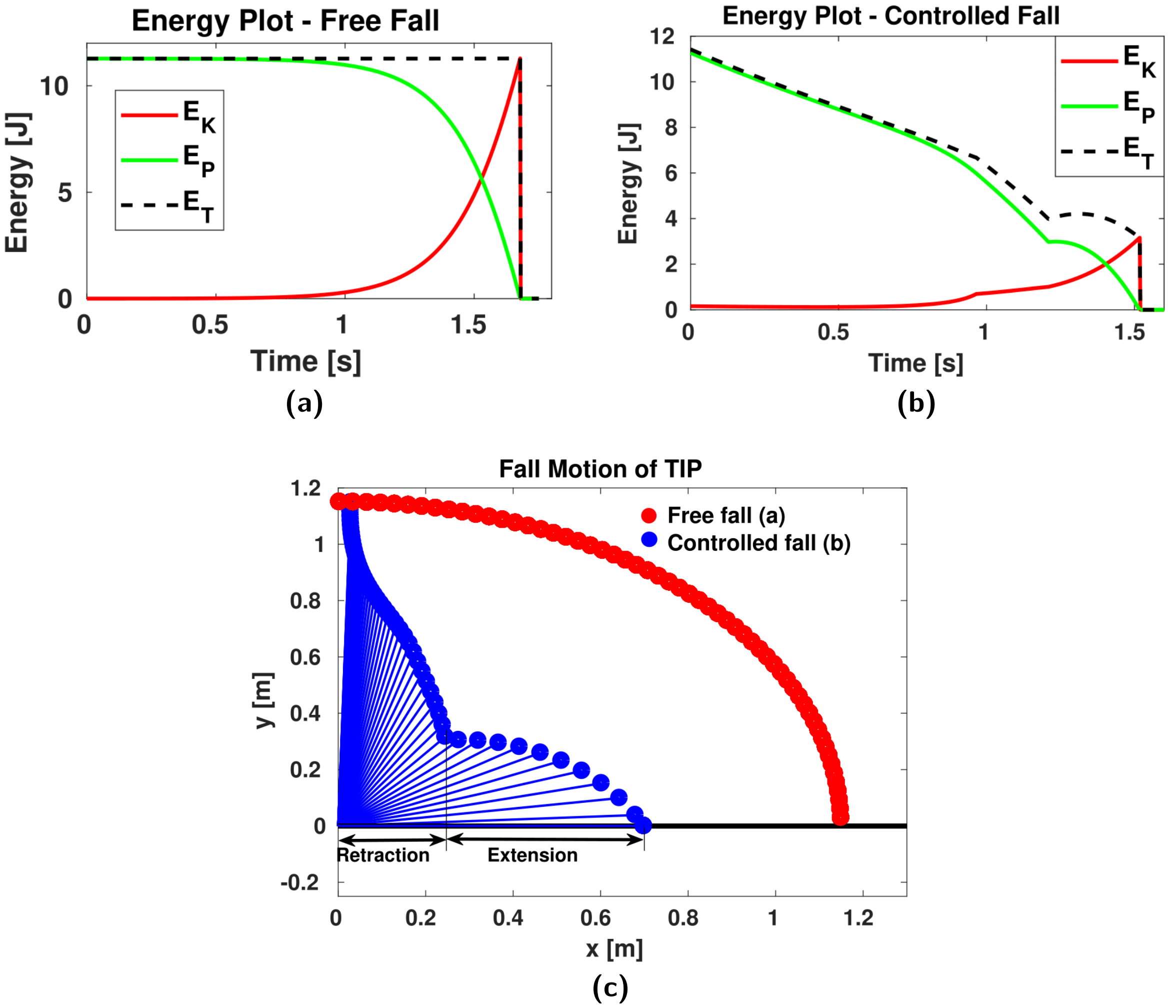


Figure 11-2: Energy plot of TIP model a) free fall. b) controlled fall with energy c) fall motion shaping.

11-3 Energy Distribution Polygon

Following energy shaping, to further reduce the impact force one of the effective solutions is to distribute the energy over multiple contacts. In this section, we propose a technique to construct a polygon with a set of body points and to adapt it as the robot falls over. We call this polygon as energy distribution polygon (EDP), which is constructed with the tipping point and possible multiple contact points on the robot. The following subsections explain the process for different fall cases.

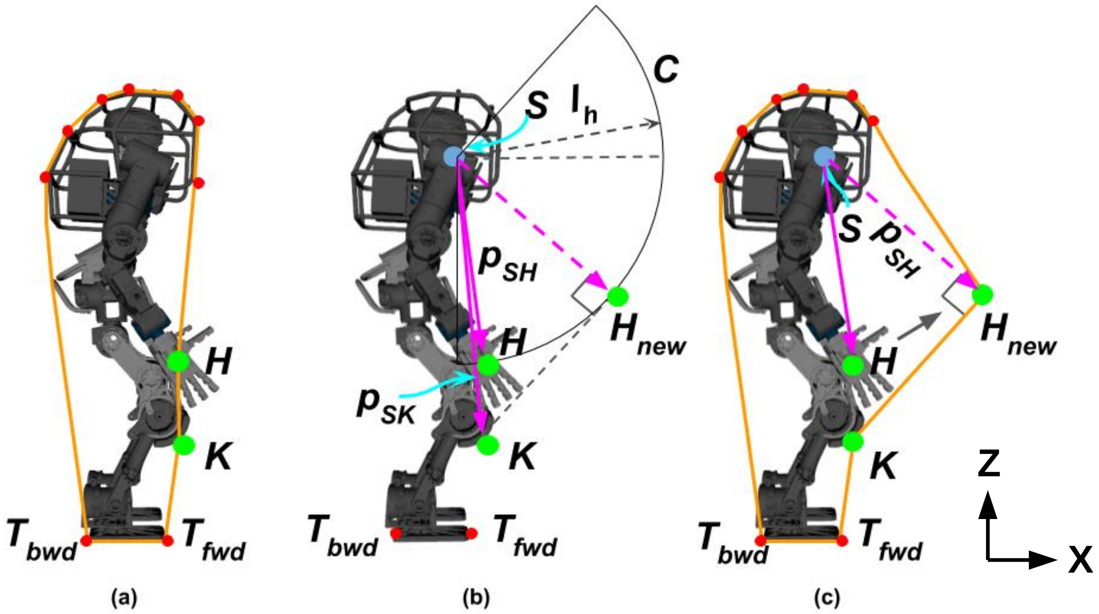


Figure 11-3: Energy distribution polygon for the forward case: (a) initial polygon state, (b) estimation of H_{new}, and (c) modified polygon with updated H location. H, K and S represents the position of hand, knee and shoulder respectively. T_{fwd} and T_{bwd} represents the forward and backward contact points of the feet.

11-3-1 Forward Fall

Figure 11-3 illustrates how EDP is constructed and modified analytically during the forward fall. In Fig. 11-3a, the green points K and H are the ones which are to hit the ground early and hence they are considered here.

The contact K is taken to be the primary contact since it is relatively close to the ground compared to H and it is already modified due to the squatting motion generated by energy shaping. Hence, only H is considered in the modification of EDP, where x and z coordinates of the point H are modified since changing its y coordinate will have a relatively less desirable effect on energy distribution. Note that point H is required to contact the ground as early as possible to reduce the energy conversion from E_P to E_K; for the updated EDP, the new point of H, H_{new}, is set to be fully stretched from the body. In addition, we need to ensure that H_{new} becomes the secondary contact to effectively dissipate the energy. Figure 11-3b shows the modified polygon for the robot with H moved to H_{new}.

To obtain the position of H_{new} the maximum possible hand length is taken as $l_h = a \cdot ||\mathbf{p}_{SH}||$, where $a \in \mathbb{R}^+ < 1$ is set close to 1, to avoid the kinematic singularity of the arm (e.g. $a = 0.9$ used in this paper). This forms the radius of the arc C shown in Fig. 11-3b.

H_{new} shown in Fig. 11-3b lies on C which is the tangent point on C for a line drawn from K. To ensure the aforementioned tangent condition, it is to be confirmed that $\angle_{HSK} =$

Algorithm 1 : Computation of H_{new} for forward fall

```
 1: if $||\mathbf{p}_{SK}|| < ||\mathbf{p}_{SH}||$ then
 2:     $d = ||\mathbf{p}_{SK}||$
 3: else
 4:     $l_h = a \cdot ||\mathbf{p}_{SH}||$
 5: end if
 6: $\theta_1 = \angle(\mathbf{p}_{SH}, \mathbf{p}_{SK})$
 7: $\hat{\mathbf{p}}_{SH} = \text{RotY}(-\frac{\pi}{4}-\theta_1) \cdot \hat{\mathbf{p}}_{SH}$
 8: $\theta_2 = \angle(\mathbf{p}_{HS}, \mathbf{p}_{HK})$
 9: if $|\theta_2-\frac{\pi}{2}| > 0.0$ then
10:     $\hat{\mathbf{p}}_{SH} = \text{RotY}(\theta_2-\frac{\pi}{2}) \cdot \hat{\mathbf{p}}_{SH}$
11: end if
12: $\mathbf{p}_{SH} = \hat{\mathbf{p}}_{SH} \cdot l_h \rightarrow \mathbf{p}_H = \mathbf{p}_S + \mathbf{p}_{SH} \rightarrow \mathbf{p}_{H_{new}} = \mathbf{p}_H$
```

$\pi/4$ and $\angle_{SHK} = \pi/2$, where $\angle_{HSK}$ is the angle between $\mathbf{p}_{SH}$ and $\mathbf{p}_{SK}$, and $\angle_{SHK}$ is the angle between $\mathbf{p}_{HS}$ and $\mathbf{p}_{KH}$. Algorithm 1 shows the pseudo-code for computing H_{new}. Line 1-5 in Algorithm 1 ensures the primary contact of K and the remaining lines illustrates the computation of H_{new}. It can be estimated by constructing two vectors $\mathbf{p}_{SK}$ and $\mathbf{p}_{SH}$, with θ_1 being the angle between them. Starting with $\theta_1 = 0.7854$ rad, H is moved along C until the perpendicularity between SH and KH is satisfied (i.e., tangency). In the algorithm, $\mathbf{p}_{SK}$ represents a vector between S and K and $\mathbf{p}_{SH}$ is between S and H, $\hat{\mathbf{p}}_{SH}$ denotes the normalized vector between S and H, $\angle(\mathbf{p_{SK}}, \mathbf{p_{SH}})$ denotes the computation of planar angle between $\mathbf{p}_{SK}$, and $\mathbf{p}_{SH}$ and RotY(θ) is the rotation matrix along Y axis with θ as angle in radians.

11-3-2 Backward and Side Fall Cases

Backward Fall:

The same principle is applied for the backward fall and the resulting EDP is shown in Fig. 11-4a. Between the possible points U and H, the point H is chosen for the following two reasons: relative closeness to the ground than U; and being part of the less inertial segments, i.e., arms, it can be controlled quickly than U which is part of the torso (high inertia). Since H is the primary contact, $\mathbf{p}_{SH}$ and $\mathbf{p}_{ST_{bwd}}$ are considered to estimate H_{new}.

Side Fall:

In the case of side fall as shown in Fig. 11-4b the possible controllable points are K, H and J. The points K and H are considered to modify the EDP due to their closeness to the ground. Similar to the forward case, K is the primary contact point (due to energy shaping) before H. $\mathbf{p}_{SH}$ and $\mathbf{p}_{SK}$ are thus considered to estimate the position of H_{new}. Note

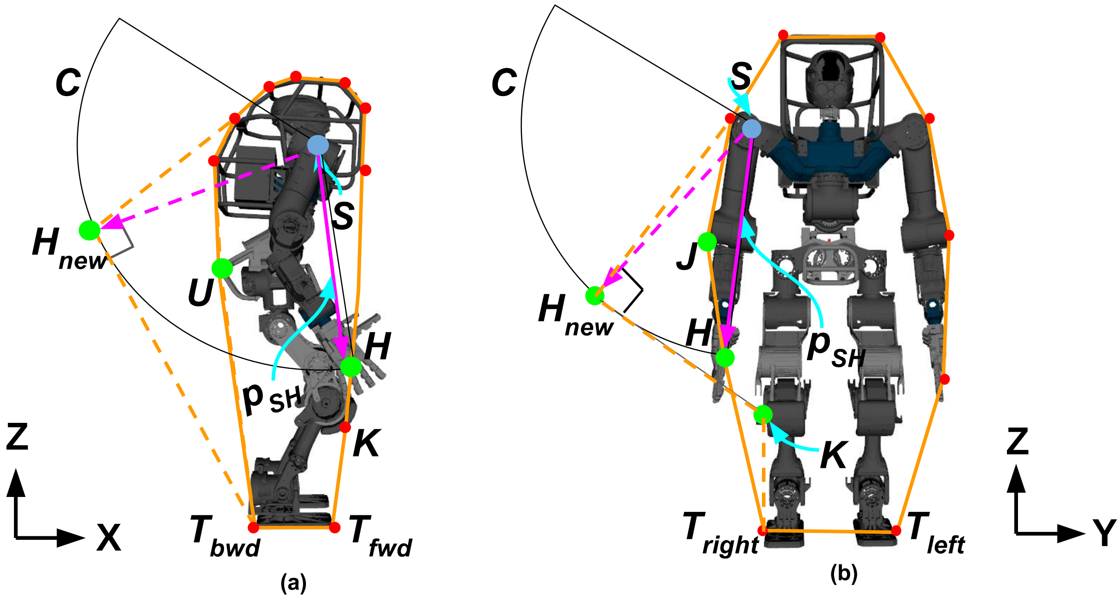


Figure 11-4: Initial (solid line) and modified (dotted line) EDPs: a) backward fall case, and b) right side fall case. U and J represents the probable contact positions of the base frame (backward fall) and right elbow (right side fall). T_{right} and T_{left} represents the side contact points of the right and left foot respectively.

that H_{new} is computed for the right hand in the case of right side fall and vice versa. In addition, y and z coordinates of the point H are modified since changing its x coordinate will have a relatively less desirable effect on energy distribution.

11-4 Whole Body Control Scheme

This section briefly explains the online whole body falling over controller developed with the concepts proposed in Sections 11-2 and 11-3. The overview of the proposed control is illustrated in Fig. 11-5. This includes the fall control module which consists of two hierarchical priority tasks:

- Task 1: energy shaping → CoG motion
- Task 2: energy distribution → end-effector motion

For Task 1, the TIP model is used to compute the control input $\dot{r}_d$ to shape the energy as discussed in Section 11-2, following which in Task 2 the minimized energy is distributed using EDP discussed in Section 11-3. The desired whole body motion is realized with the complete model of the robot in position and orientation control. The specific details of each task are given in the following subsections.

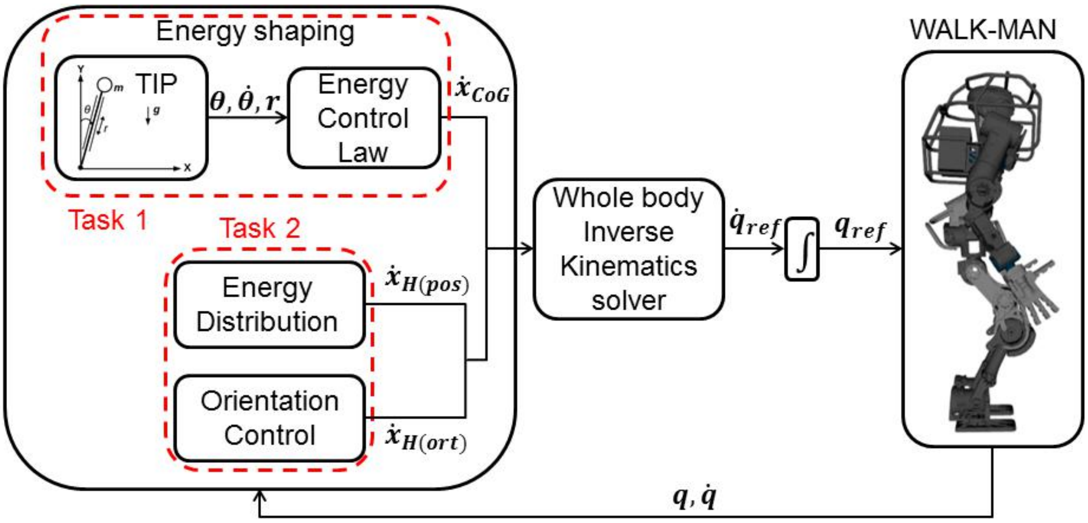


Figure 11-5: Proposed online whole body control scheme for humanoids.

11-4-1 Energy Shaping (Task 1: ES)

This task is associated with minimizing the energy using energy shaping technique. The control input $\dot{r}_d$ at each time instant is computed as explained in Section 11-2-1. The input $\dot{r}_d$ here is the desired velocity of CoG ($\dot{\mathbf{x}}_{CoG}$) in the z direction making $\dot{\mathbf{x}}_{CoG} = [0\ 0\ \dot{r}_d]^T$. The corresponding joint velocity reference ($\dot{\mathbf{q}}_{T1}$) for Task 1 can be computed as follows:

$$\dot{\mathbf{q}}_{T1} = \begin{bmatrix} \mathbf{J}_C \\ \mathbf{J}_{CoG} \end{bmatrix}^+ \begin{bmatrix} \mathbf{0} \\ \dot{\mathbf{x}}_{CoG} \end{bmatrix}, \tag{11-9}$$

where $\mathbf{J}_C \in \mathbb{R}^{n_c \times n}$ refers to the constraint Jacobian matrix which includes both the legs $(6 \times 2)$and the Z axis rotation of the torso, making the number of constraints $n_c = 13$ and the total degrees of freedom (DoF) of the robot is taken to be $n = 37$, including 6 DoF of the floating base; and $\mathbf{J}_{CoG} = \delta\mathbf{x}_{CoG}/\delta\mathbf{q} \in \mathbb{R}^{3\times n}$ represents the Jacobian of CoG, where $\mathbf{q} \in \mathbb{R}^n$ denotes the joint position vector, $\mathbf{0} \in \mathbb{R}^{n_c}$ is the zero velocity vector for the constraints and $\bullet^+$ denotes the Moore-Penrose generalized inverse of the matrix of $\bullet$.

11-4-2 Energy Distribution (Task 2: ED)

The EDP of the humanoid is modified to distribute energy among the potential contacts, i.e., end-effectors as discussed in Section 11-3. This is done by controlling the selected end-effectors to their respective optimal (locally) location ($\mathbf{x}_{Pref}$) calculated online.

The joint velocity references for Task 2 computed as follows:

$$\dot{\mathbf{q}}_{T2} = \mathbf{N}_1\mathbf{J}_{eff}^+ \left(\mathbf{K}_p\mathbf{x}_{Perr} + \mathbf{K}_d\dot{\mathbf{x}}_{Perr}\right), \tag{11-10}$$

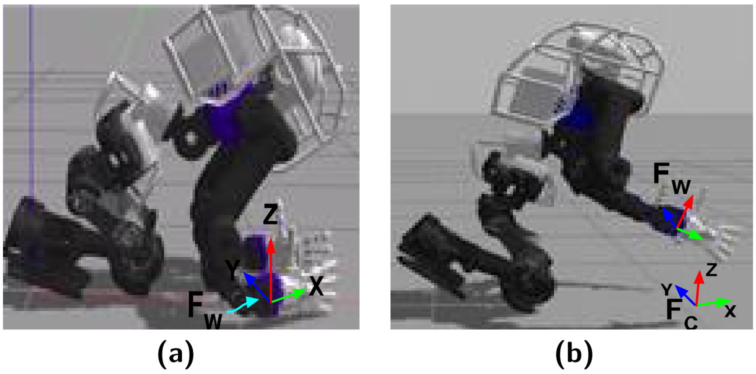


Figure 11-6: End-effector orientation control to align the wrist joint frame, $\mathbf{F_W}$ to the contact frame, $\mathbf{F_C}$.

where $\mathbf{N}_1 \in \mathbb{R}^{n \times n}$ denotes the nullspace projection matrix given by $\mathbf{N}_1 = \mathbf{I} - \mathbf{J}_A^+ \mathbf{J}_A$, where $\mathbf{I} \in \mathbb{R}^{n \times n}$ is the identity matrix and $\mathbf{J}_A$ is given by an augmented Jacobian for Task 1 as

$$\mathbf{J}_A = \begin{bmatrix} \mathbf{J}_C^T & \mathbf{J}_{CoG}^T \end{bmatrix}^T, \tag{11-11}$$

and $\mathbf{J}_{eff} \in \mathbb{R}^{(n_{eff} \cdot 3) \times n}$ is the Jacobian of n_{eff} end-effectors, while $\mathbf{K}_p$ and $\mathbf{K}_d$ as the gain matrices for the closed-loop inverse kinematics resolution, and $\mathbf{x}_{Perr} \triangleq \mathbf{x}_{Pref} - \mathbf{x}_P$ denotes the end-effector position error.

11-4-3 End-effector Orientation Control

Though EDP can distribute the energy just by controlling the position of the end-effectors, we include orientation control in addition to safeguard the hands. By orienting the hands from any arbitrary configuration to a relatively stronger (structurally) pose as shown in Fig. 11-6a ensures less damage to the hands and corresponding wrist joints. This can be achieved by orienting the local frame of the wrist joint ($\mathbf{F_W}$) to the contact frame ($\mathbf{F_C}$) as shown in Fig. 11-6b. Since the contact is always with the flat ground, $\mathbf{F_C}$ can be taken equivalent to the global frame. Including the above damage minimizing orientation control in Task 2, $\dot{\mathbf{q}}_{T2}$ will become the following:

$$\dot{\mathbf{q}}_{T2} = \mathbf{N}_1 \mathbf{J}_{eff}^+ \left(\mathbf{K}_p \begin{bmatrix} \mathbf{x}_{Perr} \\ \mathbf{x}_{Oerr} \end{bmatrix} + \mathbf{K}_d \begin{bmatrix} \dot{\mathbf{x}}_{Perr} \\ \dot{\mathbf{x}}_{Oerr} \end{bmatrix} \right), \tag{11-12}$$

where $\mathbf{x}_{Oerr}$ and $\dot{\mathbf{x}}_{Oerr}$ are the orientation error and angular velocity error respectively and $\mathbf{J}_{eff} \in \mathbb{R}^{(n_{eff} \cdot 6) \times n}$.

The final joint reference applied at time instant $(t + \Delta t)$ can be computed as follows:

$$\mathbf{q}_{ref}(t + \Delta t) = \mathbf{q}_{ref}(t) + \dot{\mathbf{q}}_{ref}(t) \Delta t, \tag{11-13}$$

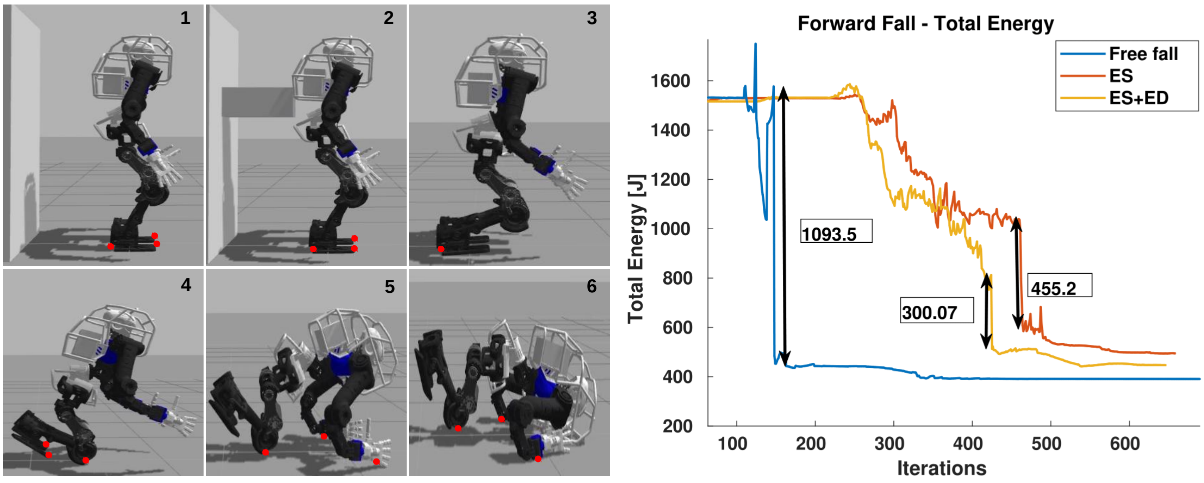
$$\dot{\mathbf{q}}_{ref} = \mathbf{S}\left(\dot{\mathbf{q}}_{T1} + \dot{\mathbf{q}}_{T2}\right), \tag{11-14}$$

where $\mathbf{S} = \left[\begin{array}{cc} \mathbf{0}_{(n-6)\times 6} & \mathbf{I}_{(n-6)\times(n-6)} \end{array}\right]$ with $\mathbf{0}$ and $\mathbf{I}$ being zero and identity matrices, to select the actuated joints.

11-5 Simulations and Results

For numerical evaluation of the proposed controller, dynamic model of the WALK-MAN humanoid (31 DoF, compliant joints, height:1.91 m, weight:140 kg) [67] has been considered. In the simulation, a pushing bar mechanism is used to exert desired forces at different locations on the robot to force it fallen over, and the fall is predicted by the method developed in [68]. For simulations, Gazebo(7.8.1) simulator with open dynamics engine (ODE) physics engine installed on a powerful workstation (2×8 cores, 32.0GB RAM and 3.8GHz processor speed) was used.

Four major falling directions forward, backward, left and right side are considered to evaluate the performance of the proposed controller. For each direction of falling over, comparative results are analyzed with only energy shaping (ES), and energy shaping with energy distribution (ES+ED). For simulation, the gains $\mathbf{K}_p = 17.5\mathbf{I}$ and $\mathbf{K}_d = 0.5\mathbf{I}$, and the threshold $\dot{r}_{max} = 3\,\text{m/s}$ are tuned to achieve better performance.



(a) (b)

Figure 11-7: Simulation results for forward fall direction is shown along with the total energy plots obtained with three different type of controllers: free fall, only ES, and ES+ED (proposed). The fall over snapshots are shown on the left side and its corresponding energy evolution to the right side. Red dots denote the contacts made with the ground during the fall over course.

11-5-1 Energy Comparison

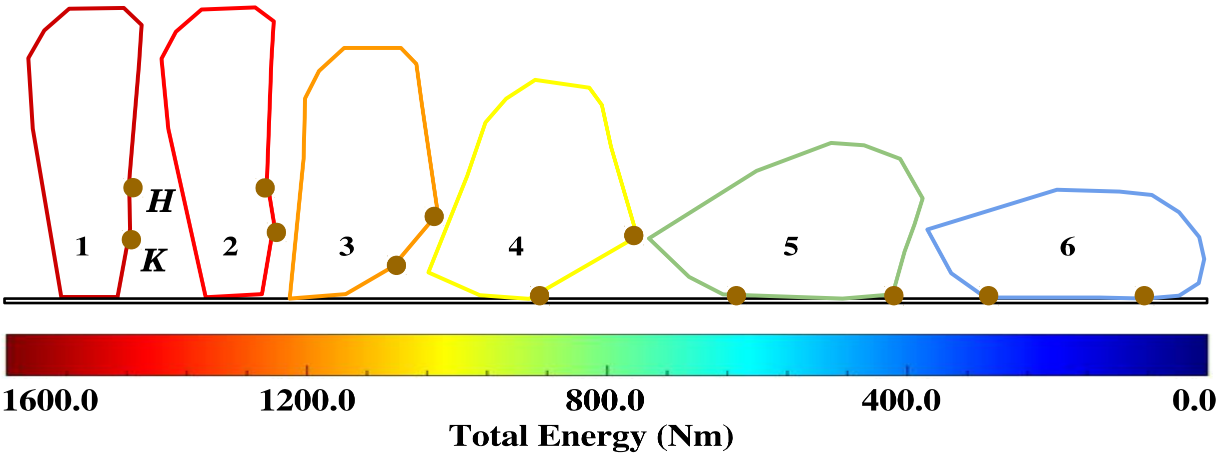


Figure 11-8: EDP for the respective forward fall snapshots shown in Fig. 11-7a, where H and K represent the hand and knee points respectively.

Forward Fall

The robot is subjected to a sequence of disturbances starting from 0 to 1100 N, applied for 50 ms at 0.9 m height from behind. Upon significant disturbance (1000 N), the robot was predicted to fall, and the proposed fall controller is activated.

In Fig. 11-7a, snapshots which were taken during the controlled fall (ES+ED) simulation are shown. Snapshot 3 shows the robot crouching because of ES and also extends the arm forward due to ED. While 4 shows robot's primary contact with its knees, followed by the secondary contact with hands in 5, and finally, in 6 because of momentum it makes contact with head protector while maintaining its hand contacts. Minimal extension motion is observed in snapshot 5, unlike the TIP model. This is due to the dissipation of a significant amount of energy through knee contact in 4. The corresponding EDP for each snapshot is shown in Fig.11-8, where H and K indicates the hand and knee contact points.

The figure clearly shows how both the tasks together reduce and distribute the energy. The plot in Fig. 11-7b shows the evolution of energy for different types of fall along with their maximum rate of energy dissipation (ΔE). Freefall having maximum ΔE = 1093.5 J is represented as Free(1093.5). ΔE observed during free fall reduces considerably with ES(455.2) because of energy shaping and primary contact of the knee, followed by ES+ED(300.07) due to the distribution of energy over additional hand contacts.

11-6 End-effector Contact Comparison

Figure 11-9 shows the effect of orientation control in Task 2 for all three fall cases. In the forward (Fig 11-9a) and backward (Fig. 11-9b) fall, with ES the hands remain above the ground at their initial position (w.r.t. the base), whereas with ES+ED they are moved to their respective local optimal position with a favorable orientation to reduce any local damages to the hands and their corresponding wrist.

With the left-side fall shown in Fig.11-9c, in the case of ES, left-hand fingers come in direct contact with the ground, while with ES+ED the contact is made with the wrist part of the hand due to the orientation control, resulting in a relatively safer arm configuration.

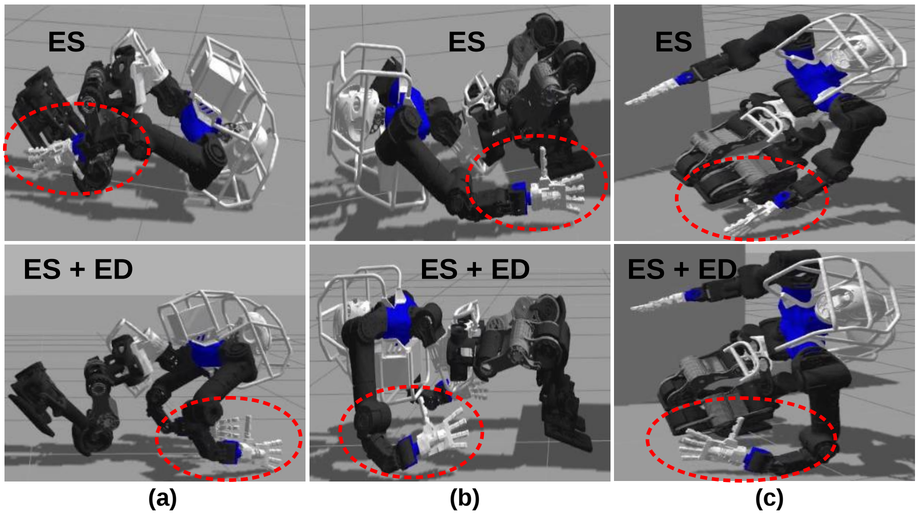


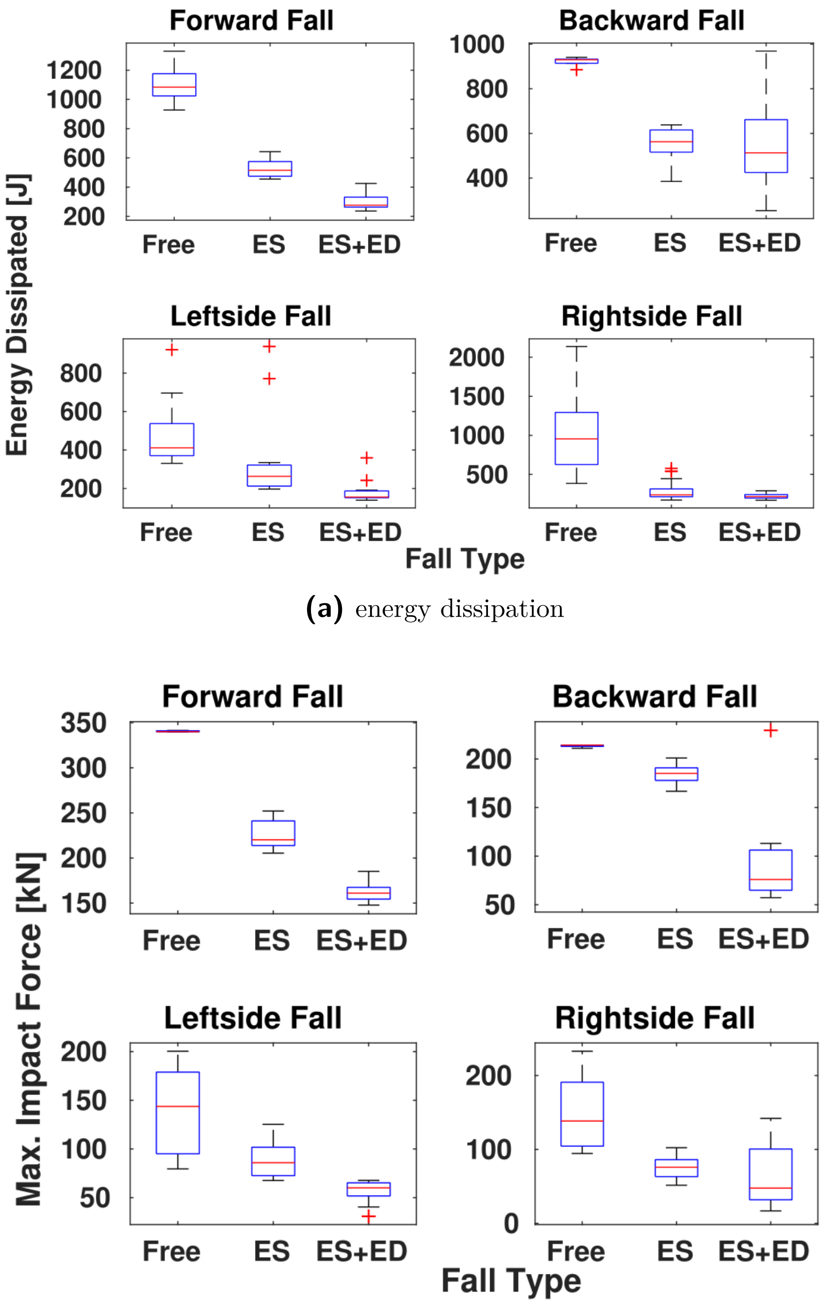
Figure 11-9: End-effector contact comparison between ES and ES+ED type controlled fall for different directions: (a) Forward, (b) backward, and (c) left-side.

11-7 Energy Dissipation and Impact Force Comparison

As explained in Section 10-1, $\mathrm{F_{imp}} \propto \Delta\mathrm{E}$ and $\mathrm{F_{imp}} \propto 1/d$. $\mathrm{F_{imp}}$ can vary due to the robot's joint compliance, disturbance, simulator (physics engine, contact sensors), etc., and hence 10 data sets are collected for each type of fall in different directions and a statistical plot is presented below in Fig. 11-10 for instantaneous dissipation of energy (ΔE) and maximum impact force.

11-7-1 ΔE Comparison

The statistical plot for energy dissipation is shown in Fig. 11-10a, which is obtained with the data collected by performing three different type of fall overs, each for 10 times, in four



(a) energy dissipation

(b) maximum impact force

Figure 11-10: Statistical plot involving 10 trials collected from three different fall cases are shown here for four major fall directions.

major directions. The instantaneous dissipation of energy (ΔE) is found to be decreasing in the following order: Free $\gg$ ES $>$ ES+ED. In the case of backward falls, the difference in ΔE between ES and ES+ED is relatively less when compared to those of forward fall overs.

This is due to the fact that the robot makes its primary contact with the hip and hands during ES and ES+ED respectively. This is unlike the latter case, where the knees make the primary contact reducing the energy conversion. Also, in side fall overs, the difference between ES and ES+ED is relatively less because in both the cases after primary contact with either the left or right knee, the robot falls over one of its arms. However, in ES+ED type control, the arm is controlled to be in a specific configuration resulting in immediate secondary contact after the knee which contributes to the reduction of ΔE.

11-7-2 F_{imp} Comparison

Overall the impact force is observed to reduce in the following order: Free $\gg$ ES $>$ ES+ED, following the same trend as observed with ΔE. This is due to the direct proportionality relationship between $\mathrm{F_{imp}}$ and ΔE. However, the reduction in $\mathrm{F_{imp}}$ is not proportional to ΔE, because of varied active joint compliance of the robot and this in turn modifies deformation (d). With the backward case, reduction in force between free and ES type fall is relatively less when compared to the forward, for the same reason stated during the energy comparison. In the case of side falls considerable variation in $\mathrm{F_{imp}}$ is observed during Free type fall. This could be associated with the robot's joint compliance since its primary contact is predominantly with the robot's upper arm, which is relatively more compliant.

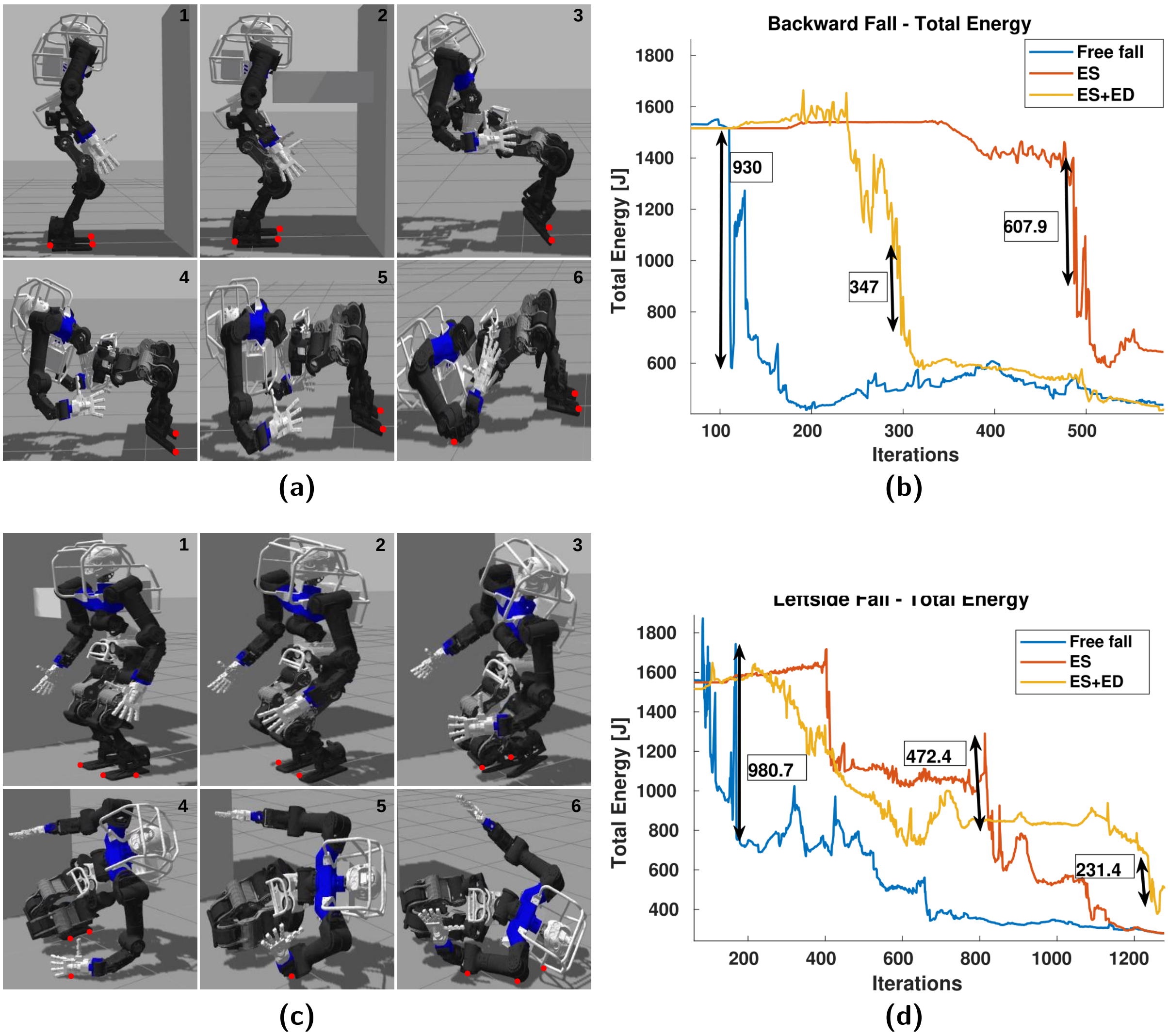
11-8 Other Direction Falls

11-8-1 Backward Fall

Disturbance similar to the forward case shown in Section 11-5 is applied but to the front side of the robot and the controlled falls are repeated.

Snapshots of ES+ED type fall are shown in Fig. 11-11a. As observed in the previous case, crouching motion of CoG and backward motion of hands were observed in snapshots 3 and 4. Unlike the forward case, primary contact is made with forearms in 5 followed by slight rolling motion resulting in the contact of the elbow and the protector as shown in 6. Also, in snapshot 5 we can notice torso straightening representing the extension motion of the robot. This is due to the absence of a contact closer to the ground, like K in the forward fall and hence ES compensates it by the generation of extension motion to reduce E_K.

Similar to the forward case, ΔE decreases in the following order Free(930)$>$ ES(667.9)$\gg$ ES+ED(347), shown in the plot in Fig. 11-11b. However, the difference between free fall and ES is relatively low because, unlike in the forward case where primary contact is made with knee (closer to the ground), here hip becomes the first contact (relatively higher from the ground) which results in additional conversion of E_P to E_K, hence the increase in ΔE.



(a)

(b)

(c)

(d)

Figure 11-11: Simulation results for backward and side fall directions are shown along with the total energy plots obtained with three different type of controllers: free fall, only ES, and ES + ED (proposed). Each row starting from the top shows the fall over snapshots on the left side and its corresponding energy evolution to the right side. The different direction fall overs are arranged as follows: backward (top), and side (bottom). Red dots denotes the contact points.

11-8-2 Side Fall

Snapshots for the left side controlled fall (ES+ED) are shown in Fig. 11-11c. Apart from the crouching and left hand motion of the robot observed in snapshots 2 and 3, tilting of the torso in the frontal plane is seen in 4 and 5 which could be due to the extension motion of ES control. The robot makes primary contact with its left hand in 4, followed by slight rolling motion it makes through successive contact with its elbow, shoulder, and head protector.

The energy dissipation among different types of fall shown in Fig. 11-11d is observed as Free(980.7) $\gg$ ES(472.4) $>$ ES+ED(231.4).

Chapter 12

Rolling Controller

The ES+ED type controller proposed in Chapter 11 was relatively static when compared to the dynamic rolling motion exhibited by judo players during the Ukemi motion. In this chapter, we extend our previously proposed controller to make successive multiple contacts to generate a rolling like motion. We hypothesize that such multiple contact motion can significantly reduce the energy dissipated per contact since the total energy is distributed across multiple contacts. This, in turn, will minimize the impact forces considerably as witnessed in the Ukemi motion.

12-1 Rolling Principle

Most of the present controlled fall techniques generate motions which involve less number of body/ground contacts. Though the resulting motions have been reported to reduce the impact forces, they are less dynamic, unlike the Ukemi motion shown in Fig. 12-1a. The discrepancy is mainly due to the less number of contacts and arbitrary body postures observed with the present controlled fall techniques. We notice that martial arts professional, who perform Ukemi motion, maintain continuous contact with the ground during fall by shaping their body (primarily using their arms and torso) in the form of a sphere/circle and roll over their shoulders. This motion avoids any damage to their vital parts such as the head and hip. The basic principle is to make continuous subsequent contacts with the ground to dissipate the accumulated kinetic energy gradually, resulting in a rolling motion. Though humans, in general, have muscles and compliant joints to absorb the impact during their fall over, they are still prone to major damages when compared to the people who perform Ukemi. Hence, we believe that Ukemi like rolling is paramount to reduce the impact forces significantly during the fall over of humanoids. Unlike humans, humanoids being rigid, the above motion can be realized by constructing a polyhedron

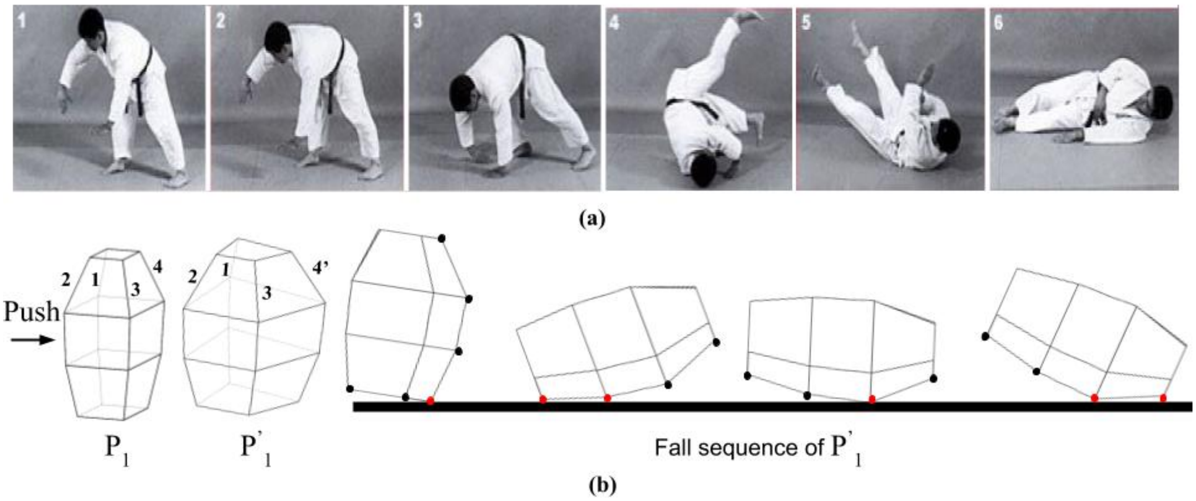


Figure 12-1: Energy distribution polyhedron to generate multi contact motion like Ukemi: (a) Ukemi forward roll motion over right shoulder, and (b) Fall over motion with the initial (P_1) and modified ($P_1^{'}$) polyhedron.

(i.e., convex hull) with the potential contacts selected around its body. The construction of the polyhedron will be discussed elaborately in Section 12-4.

12-2 Rolling Problem

In this section, we discuss what is required to realize the rolling motion with falling over humanoids, and the necessary models to do so. The insights on rolling which we made in Section 10-2 such as the influence energy has on rolling, the condition to be maintained for successive contacts, and the significance of rolling in reference to humanoids fall over, discussed in the previous section are actively used here. They are first used to define certain critical parameters, following which, the effect of choosing those parameters arbitrarily on the rolling configuration is briefly touched upon.

12-2-1 Critical Rolling Parameters

As stated earlier, energy influences rolling, and to have a favorable effect, it is necessary to distribute the dissipation of energy across all the contacts. This can be attained by carefully choosing the key parameters: contact location and its attack angles. It is important to compute these parameters since they strongly determine the performance of rolling which in turn depends on the robot posture. The robots considered here are bipeds as shown in Fig.2-1 and two different representations of it are used throughout this work: segmented planar robot (SPR) model shown in Fig.2-7a, a relatively less complex when compared to

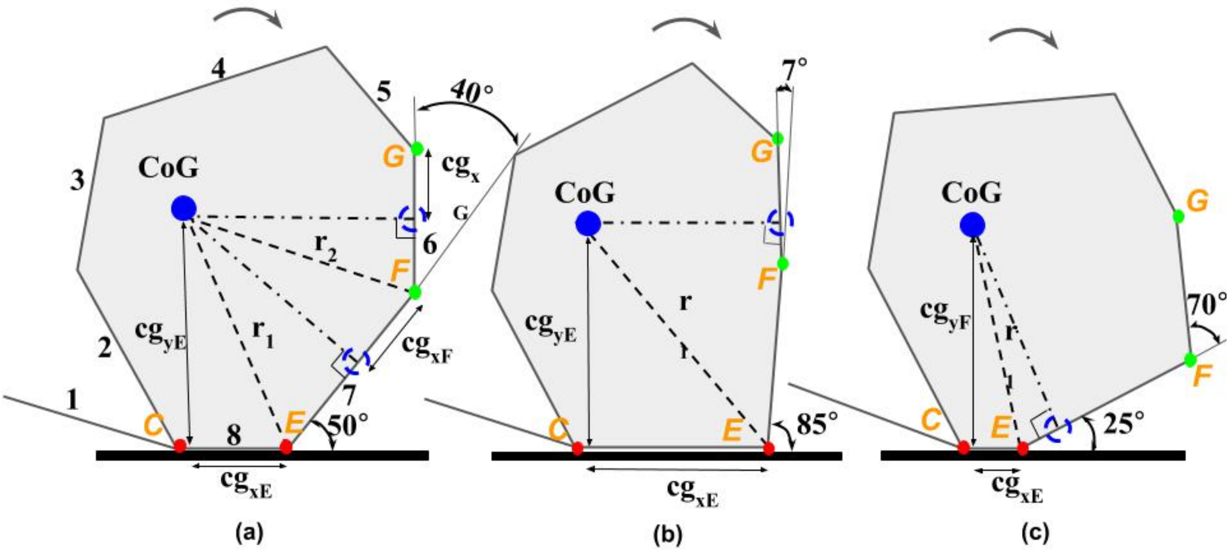


Figure 12-2: SPR in three different configurations, assuming C as its primary contact: (a) $\theta_{\text{att1}} \approx 50°$, (b) $\theta_{\text{att1}} = 85°$, and (c) $\theta_{\text{att1}} = 25°$.

the real humanoids, and the simple telescopic inverted pendulum (TIP) model shown in Fig. 2-7b.

12-2-2 How to Choose the Parameter Values?

Ideally, for a fixed e^{i-}_{ke}, the parameters have to be chosen in such a way that apart from satisfying (10-2), the work done by the system at each contact should be more or less equal. The influence of the parameters on rolling is explained analytically with the SPR model as shown in Fig. 12-2. In the figure, the blue circle represents the center of gravity (CoG) of the robot, the dotted circle denotes its projection on different links, and $cg_{\text{x}\bullet}$ and $cg_{\text{y}\bullet}$ are its relative x and y coordinates from the contact $\bullet$. The robot here is assumed to have made primary contact with C, and it is about to make its secondary contact with E. Here, the contact locations E, F, and G; the attack angles $\theta_{\text{att1}} = \pi - \angle\text{CEG}$, and $\theta_{\text{att2}} = \pi - \angle\text{EFG}$ are the parameters that need to be determined to roll over the selected contacts.

12-3 Online Rolling Controller

The optimal values for the aforementioned parameters are computed online using the controller proposed here. After predicting the inevitable fall instance of a humanoid, its state is given as input to the controller, which in turn uses the TIP and SPR models to compute the optimal values of the parameters. The versatility of SPR model in computing the critical rolling parameters for different direction fall overs is explained in Appendix A-1.

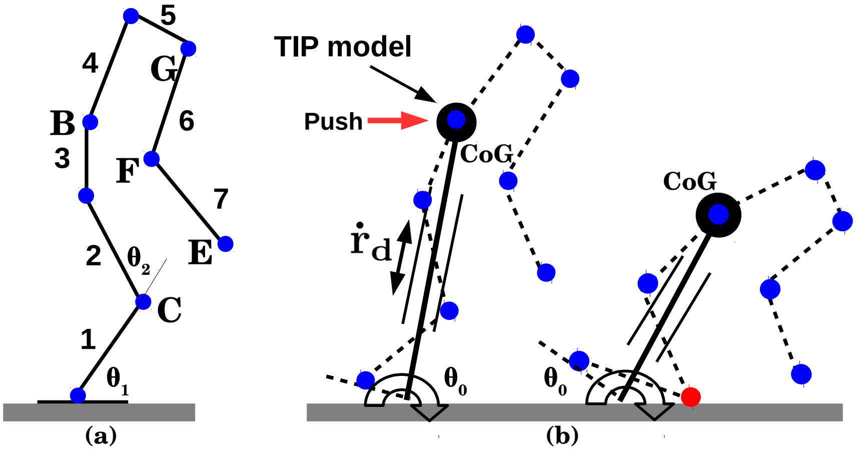


Figure 12-3: (a) SPR model of the humanoid with the contacts points knee(C), hand(E), elbow(F), shoulder(G), and waist(B) considered for constructing an optimal rolling configuration for the robot. (b)SPR and TIP models used to simulate the fall dynamics until knee (C) touches the ground.

The controller processes are shown in Fig. 12-4 and explained briefly in the following subsections.
In order to determine the optimal values of the critical rolling parameters, i.e., contact location, and attack angles, it is necessary to know the robot's state just before its primary contact. Due to the nature of external disturbances, fall prediction method, etc., the robot's state when it is about to fall can vary continuously, making it more challenging to compute the optimal values. To tackle this issue, we propose below an online rolling controller. The processes involved in the controller are explained briefly in the following subsections.

12-3-1 State Estimation at the Instant of Primary Contact

Since we intend to carry out the rolling motion after making primary contact with C, it is necessary to determine the state of the robot, i.e., velocity ($\mathbf{v}_{\mathrm{C}}^{-}$), acceleration ($\mathbf{a}_{\mathrm{C}}^{-}$) and CoG (**cg**) just before making the contact. The TIP and SPR models are used here to determine the state of the robot as shown in Fig. 12-3. The dynamics of the TIP model can be written as follows:

$$\dot{\mathbf{x}} = \mathcal{F}(\mathbf{x}, u) := \begin{bmatrix} (g\sin\theta_0 - 2\dot{r}\dot{\theta}_0)/r & \dot{\theta}_0 & u \end{bmatrix}^T, \tag{12-1}$$

where the state is defined as $\mathbf{x} = [\dot{\theta}_0\,\theta_0\,r]$, and r, θ_0, and u represent the pendulum length, tipping angle, and control input, respectively. Note that $u = \dot{r}_{\mathrm{d}}$, where $\dot{r}_{\mathrm{d}}$ is the desired sliding velocity determined by the energy shaping controller devised in [69]. This controller minimizes the total energy by lowering the robot's CoG (crouching action), which results in the primary contact C.

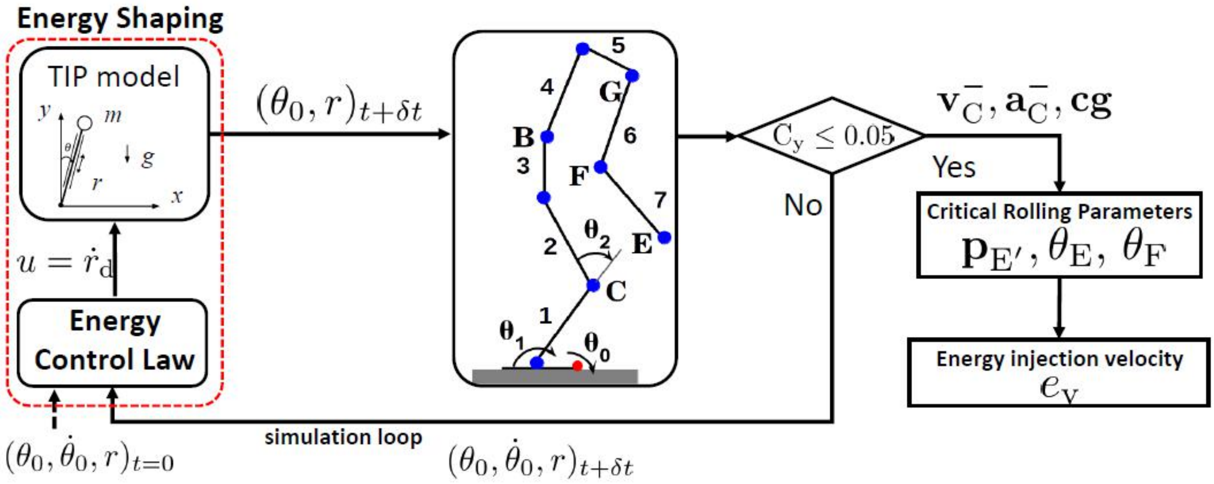

Energy Shaping
TIP model
y
m
θ
g
r
x
$u = \dot{r}_{\mathrm{d}}$
Energy Control Law
$(\theta_0, r)_{t+\delta t}$
5
4
G
B
6
3
F
7
2
θ_2
E
C
1
θ_1
θ_0
$\mathrm{C_y} \leq 0.05$
No
Yes
$\mathbf{v}_{\bar{\mathrm{C}}}, \mathbf{a}_{\bar{\mathrm{C}}}, \mathbf{cg}$
Critical Rolling Parameters
$\mathbf{p}_{\mathrm{E}'}, \theta_{\mathrm{E}}, \theta_{\mathrm{F}}$
Energy injection velocity
e_{v}
simulation loop
$(\theta_0, \dot{\theta}_0, r)_{t=0}$
$(\theta_0, \dot{\theta}_0, r)_{t+\delta t}$

Figure 12-4: The proposed Online Rolling Controller: taking the fall prediction state $(\theta_0, \dot{\theta}_0, r)$ as input and computing the parameters $\mathbf{p}_{\mathsf{E}'}, \theta_{\mathsf{E}}, \theta_{\mathsf{F}}$, and e_{v}.

Given a disturbance to the robot and upon predicting its fall, TIP model is simulated with the energy shaping control law to determine at each time step θ_0 and r as shown in Fig. 12-4. This is then sent to the SPR model to compute the new joint position of the ankle (θ_1) using inverse kinematics as,

$$\theta_1 = \sin^{-1}((\mathrm{C}_{des} - l_0 \sin(\theta_0))/l_1), \tag{12-2}$$

where $\mathrm{C}_{\mathrm{des}} = \mathrm{C}_{\mathrm{y,ini}} + (r_{\mathrm{ini}} - r)/2$ denotes the desired height of the knee, r_{ini} and $\mathrm{C}_{\mathrm{y,ini}}$ are the initial pendulum length and knee height respectively. With θ_0 and θ_1, the new knee height (C_{y}) can be given as

$$\mathrm{C}_y = l_0 \sin(\theta_0) + l_1 \sin(\theta_1 + \theta_0). \tag{12-3}$$

The above sequence is repeated until $\mathrm{C}_{\mathrm{y}} \leq 0.05m$ (knee touch down). With the above condition satisfied, the tangential velocity $(v_{\bar{\mathrm{C}}})_t = r\dot{\theta}_0$ and acceleration $(a_{\bar{\mathrm{C}}})_t = r\ddot{\theta}_0 + \dot{r}\dot{\theta}_0$ can be computed. With $(v_{\bar{\mathrm{C}}})_t$, $(a_{\bar{\mathrm{C}}})_t$, r, and θ_0 the state of the robot can be determined as

$$\mathbf{v}_{\bar{\mathrm{C}}} = v_{\mathrm{C}} \cdot \mathbf{k},\ \mathbf{a}_{\bar{\mathrm{C}}} = a_{\mathrm{C}} \cdot \mathbf{k},\ \mathbf{cg} = r_{\mathrm{C}} \cdot \mathbf{k},\ \mathbf{k} := [\sin\theta_0\ \ \cos\theta_0]. \tag{12-4}$$

12-3-2 Computation of Critical Rolling Parameters

Optimal Contact Position

In order to achieve a smooth rolling motion, it is necessary to maintain continuous contact with the ground. With C being the primary contact, E is chosen as the secondary contact due to its proximity to the ground. To prevent further conversion of the robot's potential

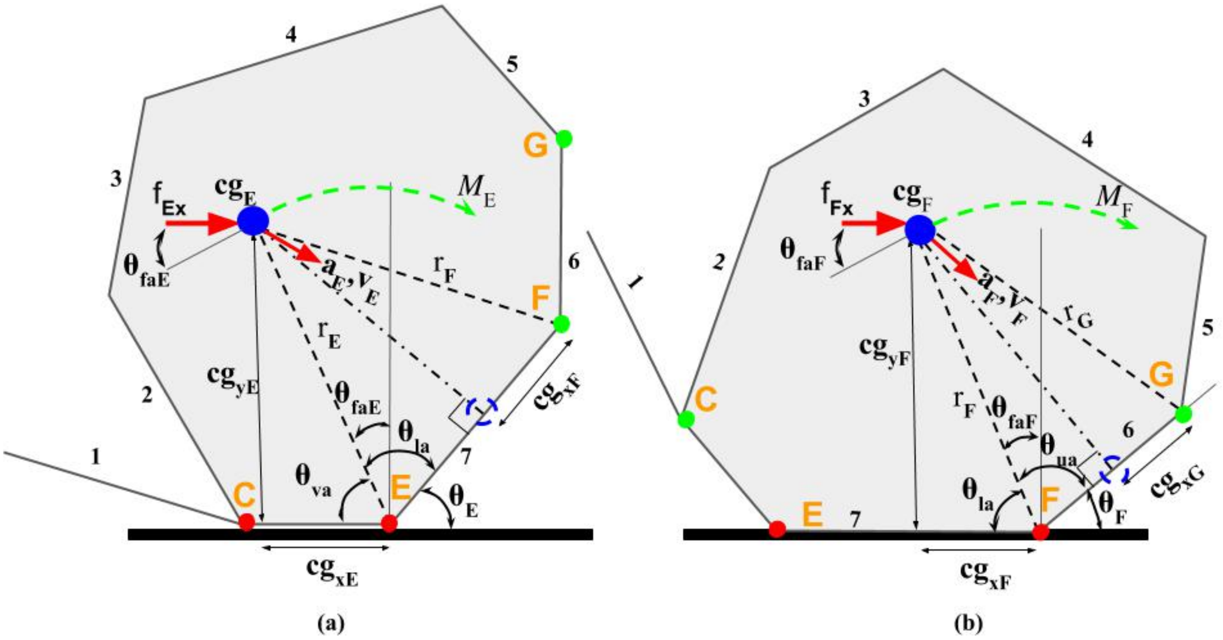


Figure 12-5: SPR represented in the form of a polygon: (a) Position and the attack angle (θ_{E}) of E with the robot state v_{E} and a_{E}), and (b) those of F with the robot state v_{F} and a_{F}.

energy (PE) to kinetic energy (KE) the z coordinate of E is maintained at the same level of C, leaving only the x coordinate of E ($\mathrm{E_x}$) to be determined as shown in Fig. 12-5a. As discussed in Section 12-2-2, $\mathrm{E_x}$ can neither be too close nor too far, hence its optimal location can be computed by equating the moment w.r.t. E and solving for cg_{xE} as

$$cg_{\mathrm{xE}} = (f_{\mathrm{Ex}} \cdot cg_{\mathrm{yE}})/m \cdot g. \tag{12-5}$$

where f_{Ex} is the x component force acting on the system after contact E (assuming E and C contacts are made simultaneously). From cg_{xE}, the optimal value of $\mathrm{E_x}$ is determined as $\mathrm{E_x} = cg_{\mathrm{x}} + cg_{\mathrm{xE}}$. Since the state of the system changes immediately after the contact due to the impact with the ground, the new state at contact E can be computed as follows:

$$(v_{\mathrm{E}}^{+})_t = \sqrt{(e_{\mathrm{ke}}^{\mathrm{E}+} \cdot 2)/m}, \quad (a_{\mathrm{E}}^{+})_t = ((v_{\mathrm{E}}^{+})_t^2 - (v_{\mathrm{E}}^{-})_t^2)/2s, \tag{12-6}$$

where $e_{\mathrm{ke}}^{\mathrm{E}+}$ and s denote the system's kinetic energy and it's rebound after contact E, respectively. They can be given as

$$e_{\mathrm{ke}}^{\mathrm{E}+} = \epsilon . e_{\mathrm{ke}}^{\mathrm{E}-}, \quad s = ((v_{\mathrm{E}}^{-})_t^2 - (v_{\mathrm{E}}^{+})_t^2)/2g. \tag{12-7}$$

where ϵ is the coefficient of restitution. For instance, if contacting parts of the robot are made of high strength aluminium alloy (7076), $\epsilon = 0.35$ [70].

Compuation of Attack Angles

The angle made by the contacts E (θ_{E}) and F (θ_{F}) are computed in this section as explained below. Since we intend a continuous rolling motion, in order to dissipate the energy gradually across the contacts, it is preferable to maintain the same tipping radius, i.e., $r_{\mathrm{E}} = r_{\mathrm{F}} = r_{\mathrm{G}}$. The tipping radius about contact E (r_{E}) can be computed from the position of E and CoG. With the length of the lower arm link $||\mathbf{p}_{\mathrm{EF}}||$, and upper arm link $||\mathbf{p}_{\mathrm{FG}}||$, θ_{E} and θ_{F} can be computed as

$$\theta_{\mathrm{E}} = \cos^{-1}((0.5||\mathbf{p}_{\mathrm{EF}}||)/r_{\mathrm{E}}), \tag{12-8}$$

$$\theta_{\mathrm{F}} = \cos^{-1}((0.5||\mathbf{p}_{\mathrm{FG}}||)/r_{\mathrm{F}}). \tag{12-9}$$

Energy Injection

A system's continuous rolling depends not only on its contact positions and attack angles but also on the *energy rolling condition* (10-2). Failing to satisfy (10-2) could hamper the rolling motion of a system, called *energy drought.* The drawback of this is the reduction of the number of contacts, over which initial kinetic energy e^{-}_{ke} is distributed, then, this relatively increases the impact force at the last contact. In this work, we propose to avoid the energy drought condition by injecting energy into the system at the beginning of the rolling motion. The energy injection process is briefly explained below.

With the robot's state —$\mathbf{v}^{-}_{\mathrm{E}}$, $\mathbf{a}^{-}_{\mathrm{E}}$, and **cg**—just before making the contact C/E known from simulating the TIP model, the initial kinetic energy can be computed as $e^{\mathrm{E}-}_{\mathrm{ke}} = 0.5m\cdot(v^{-}_{\mathrm{E}})^2$. The energy available after each impact for rolling over any contact point i can be computed as

$$e^{i}_{\mathrm{ke}} = \epsilon \cdot e^{i-}_{\mathrm{ke}}, \tag{12-10}$$

for example, at contact point E, $e^{\mathrm{E}+}_{\mathrm{ke}} = \epsilon \cdot e^{\mathrm{E}-}_{\mathrm{ke}}$. With the state after impact at E computed in (12-6), the work done in rotating the system over a radius r_{E} about E can be computed as

$$w_{\mathrm{E}} = mg\mathrm{r}_{\mathrm{E}}(1 - \cos(\theta_{\mathrm{faE}})), \tag{12-11}$$

where the right side term denotes the change in system's potential energy during the tipping motion.

Even though the total rotation of the system about E would be θ_{E} as shown in Fig 12-5a, only until θ_{faE} work is done by the system since it involves raising the CoG against the gravity, while for $(\theta_{\mathrm{E}} - \theta_{\mathrm{faE}})$ the work is done by the gravity due to the lowering of CoG. It is necessary to satisfy the condition, $e^{\mathrm{E}+}_{\mathrm{ke}} \geq w_{\mathrm{E}}$, to successfully roll over E. Similarly, the *energy rolling condition* can be verified for other contacts as well.

With the available energy and the work done computed separately for each contact, the total work done and the total available energy can be computed as $w^{\mathrm{t}} = w_{\mathrm{E}} + w_{\mathrm{F}}$ and $e^{\mathrm{t}}_{\mathrm{ke}} = e^{\mathrm{E}+}_{\mathrm{ke}} + e^{\mathrm{F}+}_{\mathrm{ke}}$. The energy to be injected at each contact ($e^{i}_{\mathrm{ke,inj}}$) is computed as

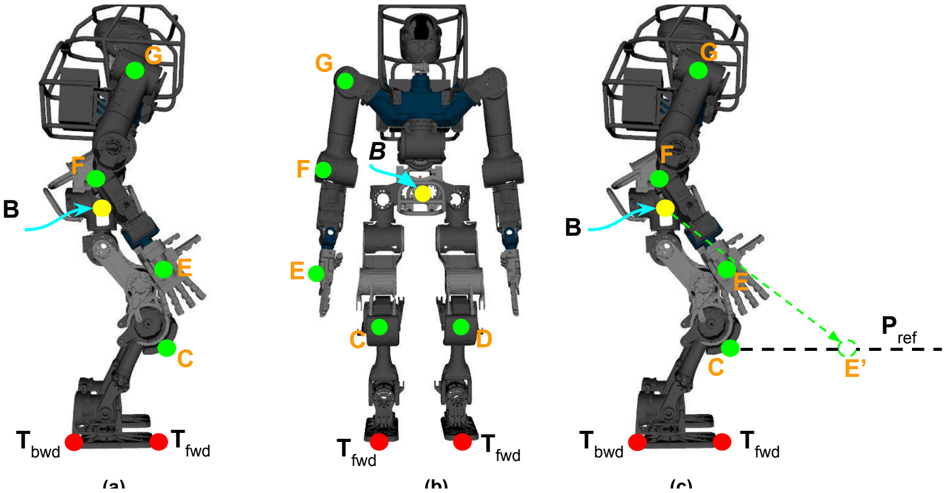


Figure 12-6: WALK-MAN with the potential contacts selected for right arm roll shown from different views: (a) right side and (b) front view. (c) Optimal position of E (E') computed using ORC along the primary reference line (P_{ref}).

$$e^i_{\text{ke,inj}} = \begin{cases} (w^i - e^i_{\text{ke}})/\epsilon^i, & \text{if } w^i > e^i_{\text{ke}} \\ 0, & \text{otherwise.} \end{cases} \tag{12-12}$$

The total energy needs to be injected can be written as $e^{\text{t}}_{\text{ke,inj}} = \sum_{i=1}^{n} e^i_{\text{ke,inj}}$. Then, the injection velocity e_{v} is obtained as

$$e_{\text{v}} = \sqrt{(2 \cdot e^{\text{t}}_{\text{ke,inj}})/m}. \tag{12-13}$$

By setting this to the system at the beginning of the controlled fall, we can realize continuous rolling motion without succumbing to *energy drought* condition.

12-4 Energy Distribution Polyhedron

In the previous section, the optimal values for the critical rolling parameters were computed using the TIP and SPR models. Those are used here to construct a polyhedron with the potential contacts selected around the humanoid's (WALK-MAN) body to generate a rolling motion. The construction process is explained briefly below for the forward fall case.

The construction of energy distribution polyhedron (EDPH) involves three major steps: selection of potential contacts and determining the suboptimal location for the secondary contact, alignment of the contacts to ensure their successive touch down on the ground, and finally, orienting the set of contacts along the falling direction. The proposed forward rolling can be done on either arm, here the right arm is considered for the explanation and Fig. 12-6 shows the potential contacts selected for it. C and D represent the left and right leg knee, and E, F, and G denote the right arm's hand, elbow, and shoulder contacts respectively. While T_{fwd} and T_{bwd} denote the forward and backward tipping point, and B represents the local base frame of the robot.

The knee contacts (C and D) are assumed to be the primary contacts due to the crouching action of energy shaping (ES). For the right shoulder roll (discussed here), the right knee contact C, in particular, is taken as a reference and the optimal location of E, i.e., E' is computed as stated in Section 12-3-2.

Following the contacts C and E', the other contacts (F and G) are aligned to ensure continuous successive touch down on the ground. This is done by first constructing a plane P1 along the primary reference line P_{ref} with E', and then forming a convex polygon with E', F and G using the link lengths $L_1 = ||\mathbf{p}_{\text{EF}}||$ and $L_2 = ||\mathbf{p}_{\text{FG}}||$. The convexity of the polygon is controlled using the attack angles (θ_{E} and θ_{F}) computed with online rolling controller (ORC) in Section 12-3-2. The constructed polygon is shown in Fig. 12-7a.

Finally, the aligned contacts are oriented in the direction of the robot's fall (f_{dir}) by orienting P1 along with the contacts towards the fall over direction, resulting in $\text{E}^{''}$, $\text{F}^{''}$, and $\text{G}^{''}$. The orientation is done with respect to the B frame by θ_{dir} along the z-axis. f_{dir} and θ_{dir} are determined and given as inputs to the proposed controller by the fall prediction method discussed in Chapter 3. The initial and final orientation of the plane are shown in Figs. 12-7b and 12-7c respectively.

12-4-1 Backward Fall

The potential contacts selected for the backward fall is shown in Fig. 12-8a. Similar to the forward fall, the rolling can be done either over left or right shoulder (depends on the fall direction), here the left arm contacts (E, F, and G) are selected to roll over the left-hand shoulder. Since during backward fall the knees cannot contact the ground, upper thigh leg contacts C and D are considered to make the primary contact with the ground. As discussed earlier, the optimal location for the secondary contact E is computed as explained in Section 12-3-2. Following which, the locations of $\text{E}^{''}$, $\text{F}^{''}$, and $\text{G}^{''}$ are computed as done earlier for the forward roll over motion. In the backward case, the robot with an onboard computer and a protector frame on its back is more likely to contact with the ground. In order to achieve a smooth contact transition over the protector frame, a convexity is introduced using the contacts H and I on the waist and backpack protector frames respectively. Since there are no active joints in the B-H chain, H is only controlled implicitly through energy shaping. With H as reference, the suboptimal position of I is located by rotating the link H-I about H in the global Y axis by θ_F, the attack angle formed by the elbow and shoulder contacts computed using ORC. P_{ref} is oriented along the negative direction of x-axis since the humanoid is falling over in that direction. Here also the values for f_{dir} and θ_{dir} are computed using the fall prediction method.

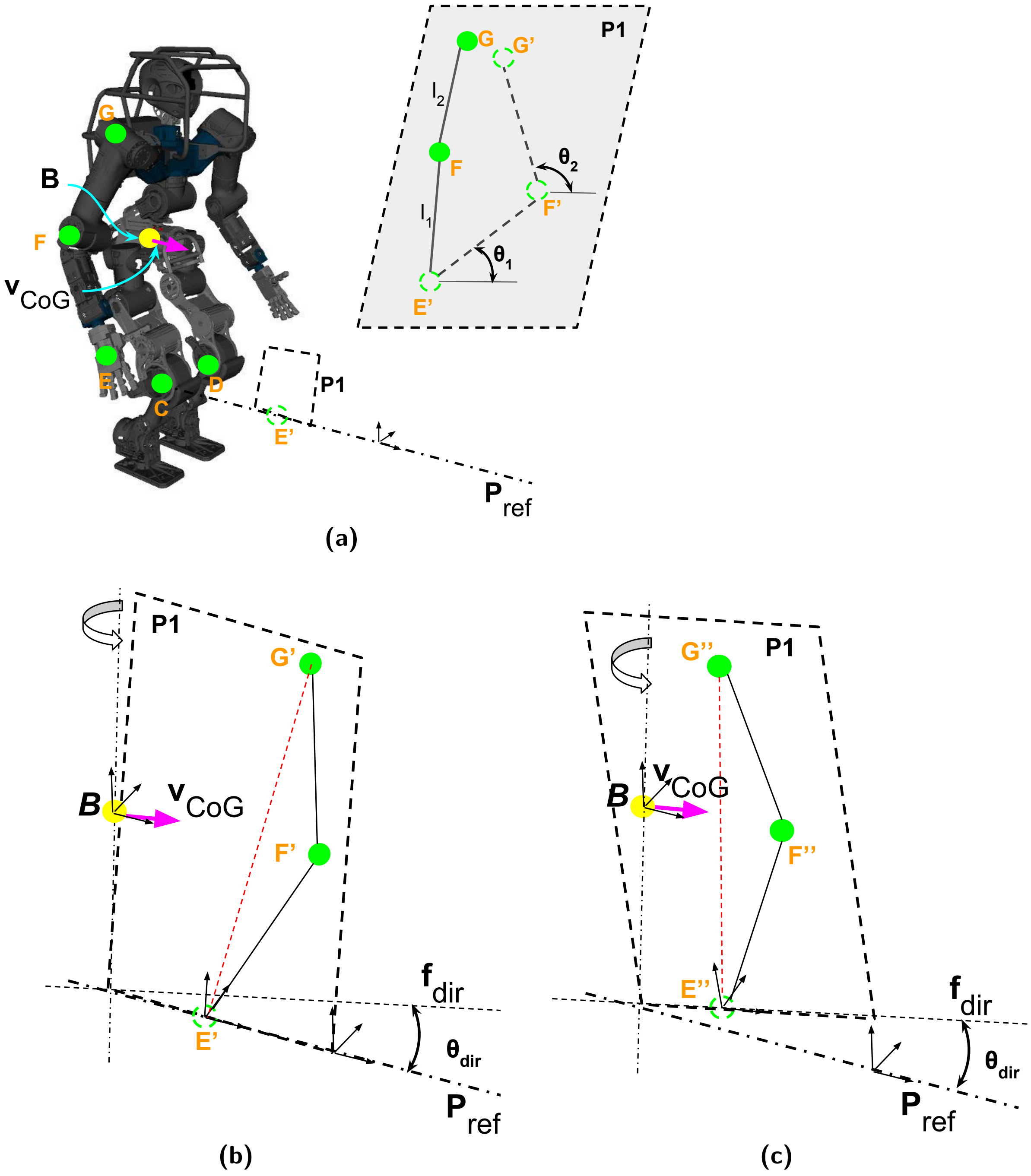


Figure 12-7: (a) Potential contacts for right side forward fall with plane P1 and the optimal position of the contacts $E^{'}$, $F^{'}$, and $G^{'}$ computed using the attack angles θ_1 and θ_2, (b) initial orientation of P1 with reference to the fall over direction axis f_{dir}, and (c) final orientation of P1 to align it along f_{dir} axis by rotating it about the base frame B by θ_{dir}.

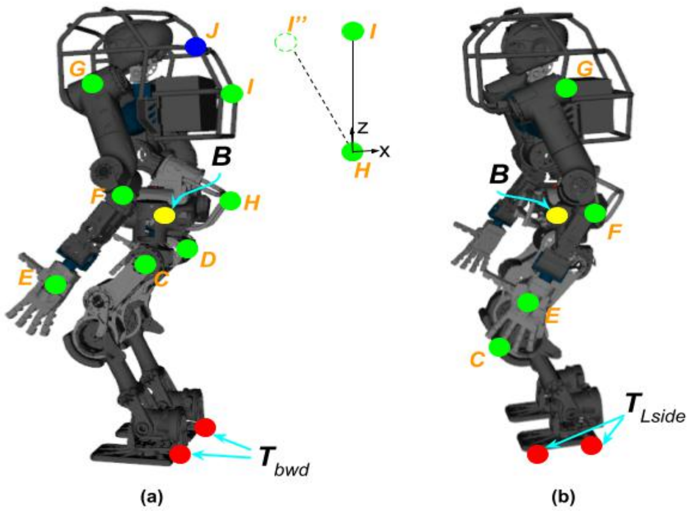


Figure 12-8: Potential contacts selected on the WALK-MAN humanoid for (a) backward and (b) left side roll during fall.

12-4-2 Side Fall

For the side fall, assuming the robot is falling to its left, the contacts C, E, F, and G are considered for generating a rolling motion as shown in Fig. 12-8b. Similar to the other cases, the leg contacts are assumed to be the primary ones due to energy shaping, and in particular, C is taken as a reference since the humanoid is falling over to its left side. The optimal locations for the selected contacts E'', F'', and G'' are computed as done for the other cases using E', θ_{E}, and θ_{F} determined with the ORC. $\mathrm{P}_{\mathrm{ref}}$ is taken along the positive y-axis since the robot is falling to its left side.

12-5 Whole Body Control for Rolling Motion

The online whole body controller presented here to achieve the proposed rolling motion includes two major tasks, and they are executed in a hierarchical order as follows:

- Task 1: energy injection and shaping → CoG commands
- Task 2: energy distribution (rolling) → pose commands

Though the tasks look similar to our controller proposed in [69], they differ considerably. The major differences in the present controller as a result of our novel contributions are observed in both the tasks mainly due to ORC. For Task 1, the TIP model is used to compute the desired control velocity $\dot{r}_{\mathrm{d}}$ to shape the energy. Additionally, here, energy injection velocity (e_{v}) computed in (12-13) is used to overcome *energy drought*, to facilitate

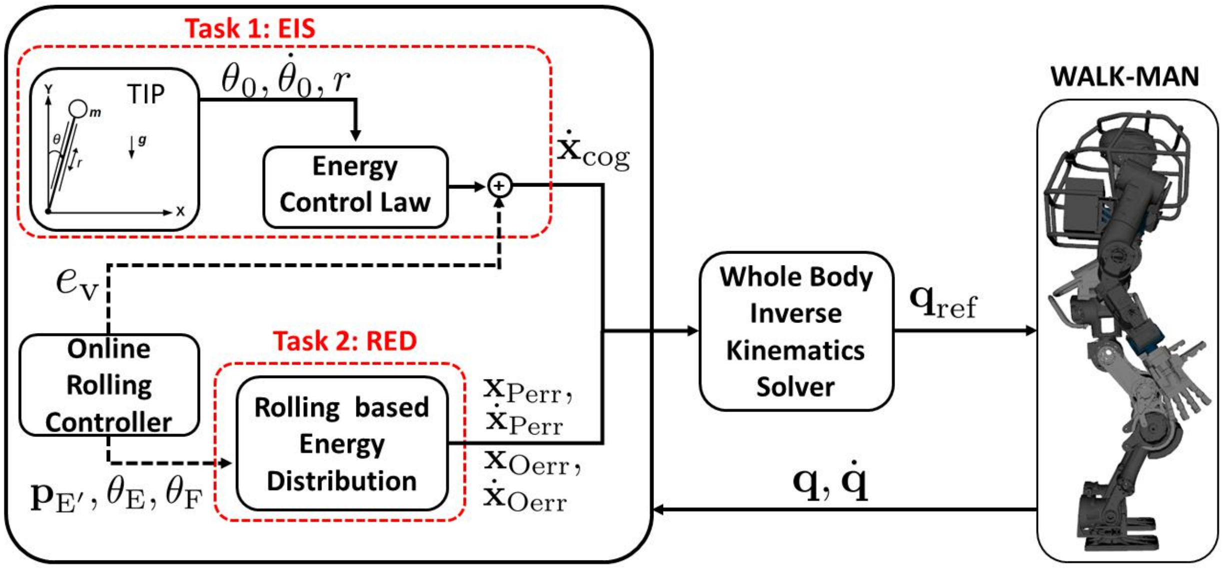

Task 1: EIS
TIP
m
θ
r
g
Y
X
$\theta_0, \dot{\theta}_0, r$
Energy Control Law
+
$\dot{\mathbf{x}}_{\text{cog}}$
e_{v}
Online Rolling Controller
Task 2: RED
Rolling based Energy Distribution
$\mathbf{x}_{\text{Perr}}, \dot{\mathbf{x}}_{\text{Perr}}$
$\mathbf{x}_{\text{Oerr}}, \dot{\mathbf{x}}_{\text{Oerr}}$
$\mathbf{p}_{\text{E}'}, \theta_{\text{E}}, \theta_{\text{F}}$
Whole Body Inverse Kinematics Solver
$\mathbf{q}_{\text{ref}}$
WALK-MAN
$\mathbf{q}, \dot{\mathbf{q}}$

Figure 12-9: Overview of the generalized online fall control for humanoids.

rolling. This is done by setting velocity commands to the movement of CoG in x and y-direction. In Task 2, the minimized energy is distributed over multiple contacts using EDPH as discussed in Section 12-4. The overview of the proposed controller is shown in Fig. 12-9.

By predicting the inevitable fall of the robot using [68], the fall controller is triggered. While the ORC computes optimal values for critical rolling parameter (CRP), Task 1 and Task 2 are performed with the default settings until an update is received from ORC. The following subsections explain briefly the details of each task.

12-5-1 Energy Injection and Shaping (Task1: EIS)

This task is associated with minimizing the energy using energy shaping technique, as introduced in [69]. However, one addition here is the inclusion of e_{v} (introduced in Section 12-3-2) to add some energy into the system to overcome *energy drought* during the multi-contact motion. The intended motion along with the energy shaping action is achieved by setting the desired velocity of CoG, $\dot{\mathbf{x}}_{\text{CoG}} = [e_{\text{vx}} + \dot{r}_{\text{d}} \cos(\theta_0) \; e_{\text{vy}} \; \dot{r}_{\text{d}} \sin(\theta_0)]$, where e_{vx} and e_{vy} represent the x and y components of e_{v} computed as $e_{\text{v}} \cos(\theta_{\text{dir}})$ and $e_{\text{v}} \sin(\theta_{\text{dir}})$. θ_{dir} represents the fall direction considered in Section 12-4, and θ_0 is the tilting angle of the humanoid. The joint velocity reference ($\dot{\mathbf{q}}_{\text{T1}}$) for Task 1 can be computed as discussed in [69].

12-5-2 Rolling Based Energy Distribution (Task2: RED)

The EDPH discussed in Section 12-4 is used to distribute the minimized energy over multiple contacts, to result in a smooth rolling like motion. This is done by selecting suitable contacts over the robot's body, and controlling them to their respective optimal location ($\mathbf{x}_{\mathrm{ref}}$) computed using the ORC. For forward falls, these contacts are primarily distributed over either left or right arm and the legs as shown in Fig. 12-6.

In addition, similar to our previous controller, where we included orientation control of hands to safeguard them, here also it is considered. Structurally stronger configuration was determined to align the X-axis of the wrist frame ($\mathrm{F_E}$) to the global frame ($\mathrm{F_G}$). But, since we intend to make rolling like motion, it is necessary to maintain a suitable orientation of the secondary contact, to ensure a smooth transition to the successive contacts to follow. The desired orientation of the secondary contact is obtained in two phases as shown in Fig. 12-10. First, is to orient the x-axis of F_E to the desired axis (x') by rotating x along the xz plane by θ_E (obtained from EDPH), shown in Fig. 12-10a. Second, is to align the Y-axis of F_E to the desired axis (Y'), defined as $RotZ(\theta_{ort}) \times \mathbf{n}_{\bullet}$, where $RotZ()$ is the rotation about z-axis and $\mathbf{n}_{\bullet}$ is the norm vector for fall direction $\bullet$. For forward fall $\mathbf{n}_{\bullet}$ can be taken as $[1\,0\,0]$. For left and right hands, $RotZ()$ is clockwise and anticlockwise respectively. This is shown in Fig. 12-10b. The final orientation of the hand for a right side shoulder roll over fall is shown in Fig. 12-10c.

The desired joint velocities for Task 2 ($\dot{\mathbf{q}}_{\mathrm{T2}}$) with the orientation control, can be written as

$$\dot{\mathbf{q}}_{\mathrm{T2}} = \mathbf{N}_1 \mathbf{J}_{\mathrm{con}}^{+} \left(\mathbf{K}_{\mathrm{p}} \begin{bmatrix} \mathbf{x}_{\mathrm{Perr}} \\ \mathbf{x}_{\mathrm{Oerr}} \end{bmatrix} + \mathbf{K}_{\mathrm{d}} \begin{bmatrix} \dot{\mathbf{x}}_{\mathrm{Perr}} \\ \dot{\mathbf{x}}_{\mathrm{Oerr}} \end{bmatrix} \right), \tag{12-14}$$

where $\mathbf{N}_1 \in \mathbb{R}^{\mathrm{n}\times\mathrm{n}}$ denotes the nullspace projection matrix of Task 1

and $\mathbf{J}_{\mathrm{con}} \in \mathbb{R}^{((6\times\mathrm{n_s})+(\mathrm{n_{con}}-\mathrm{n_s})\times 3))\times\mathrm{n}}$ is the Jacobian of $\mathrm{n_{con}}$ contacts, with $\mathrm{n_s}$ denoting the number of secondary contacts. $\mathbf{K}_{\mathrm{p}}$ and $\mathbf{K}_{\mathrm{d}}$ are the gain matrices of size $(6 \times \mathrm{n_s}) + (\mathrm{n_{con}} - \mathrm{n_s}) \times 3$, and $\mathbf{x}_{\mathrm{Perr}}$, $\dot{\mathbf{x}}_{\mathrm{Perr}}$, $\mathbf{x}_{\mathrm{Oerr}}$, and $\dot{\mathbf{x}}_{\mathrm{Oerr}}$ represent the contact pose and velocity errors. The desired motion of the robot is realized with position control, by commanding the reference joint values $\mathbf{q}_{\mathrm{ref}}$, obtained by integrating the reference velocities $\dot{\mathbf{q}}_{\mathrm{ref}}$.

12-6 Evaluation With SPR Model

In this section, the proposed whole body control scheme is evaluated with the SPR model shown in Fig.2-7a. In particular, the influence of energy injection during a controlled fall and the tracking performance of the controller are analyzed in detail.

12-6-1 Adams Model and Settings

SPR model shown in Fig.2-7a is modeled in Adams, with each link's kinematic and dynamic properties taken equivalent to those of the humanoid, WALK-MAN. Figure 12-11 shows

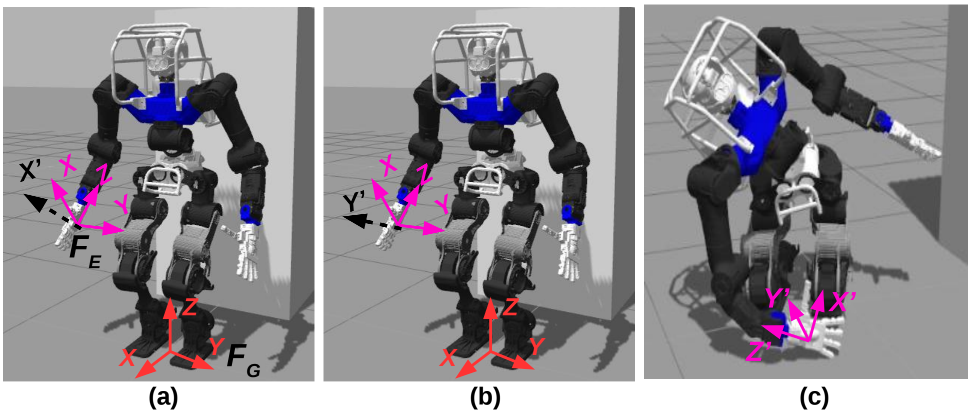


Figure 12-10: Orientation control of the hand frame $\mathsf{F_E}$ with the desired axes X' and Y'.

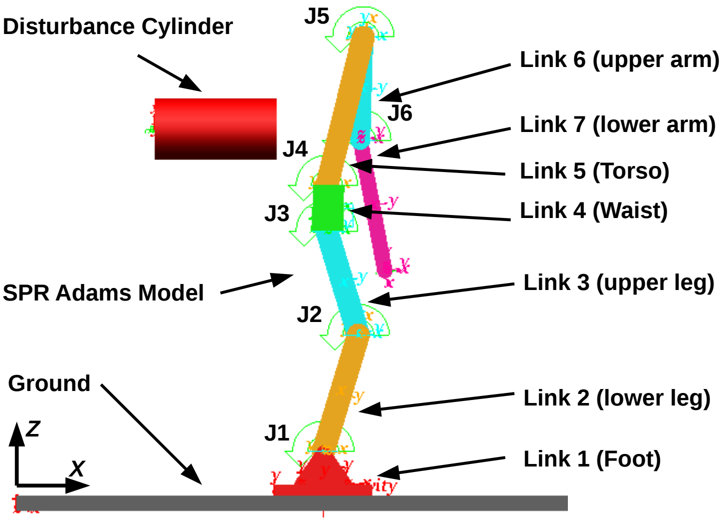


Figure 12-11: Adams simulation setup showing the SPR model, and a cylindrical block to exert disturbance on the robot. $\mathsf{J}_\bullet$ and $\mathsf{L}_\bullet$ represents the 6 active joints and the links of the model.

the Adams model, with a rectangular block representing the ground, and the cylindrical block denotes the ram used to exert disturbance on the robot. In the figure, $J_1 - J_6$ are the ankle, knee, hip, torso, shoulder and elbow joints respectively. SPR being a floating base planar model, it has 3 floating base degrees of freedom (DoF) and 6 active joints, making n = 9. Accordingly, for SPR model, the dimensions of the following variables in the control scheme are modified as follows: $\mathbf{J}_{\mathrm{C}} \in \mathbb{R}^{\mathrm{n_c}\times\mathrm{n}}$ ($\mathrm{n_c} = 3$); $\mathbf{J}_{\mathrm{CoG}} \in \mathbb{R}^{2\times\mathrm{n}}$; $\mathbf{J}_{\mathrm{con}} \in \mathbb{R}^{(\mathrm{n_{con}}\times 2)\times\mathrm{n}}$; $\mathbf{K}_{\mathrm{p}}$ and $\mathbf{K}_{\mathrm{d}} \in \mathbb{R}^{(\mathrm{n_{con}}\times 2)\times(\mathrm{n_{con}}\times 2)}$. Since no end-effector is considered in the SPR model, the orientation of the secondary contact is ignored during control. The model is controlled using the control scheme described in Section 12-5, which is implemented as a Matlab-Simulink model to communicate with the Adams simulator.

Adams solver settings:- Time step: 0.005s; contact model: type (impact), stiffness (1.0E+05 N/mm), force exponent (2.2), damping (10.0 Ns/mm), penetration depth (0.1 mm), friction type (Coulomb), static and dynamic coefficient (0.3 and 0.1 no units), stiction and friction transition velocity (100.0 mm/s and 1000.0 mm/s).

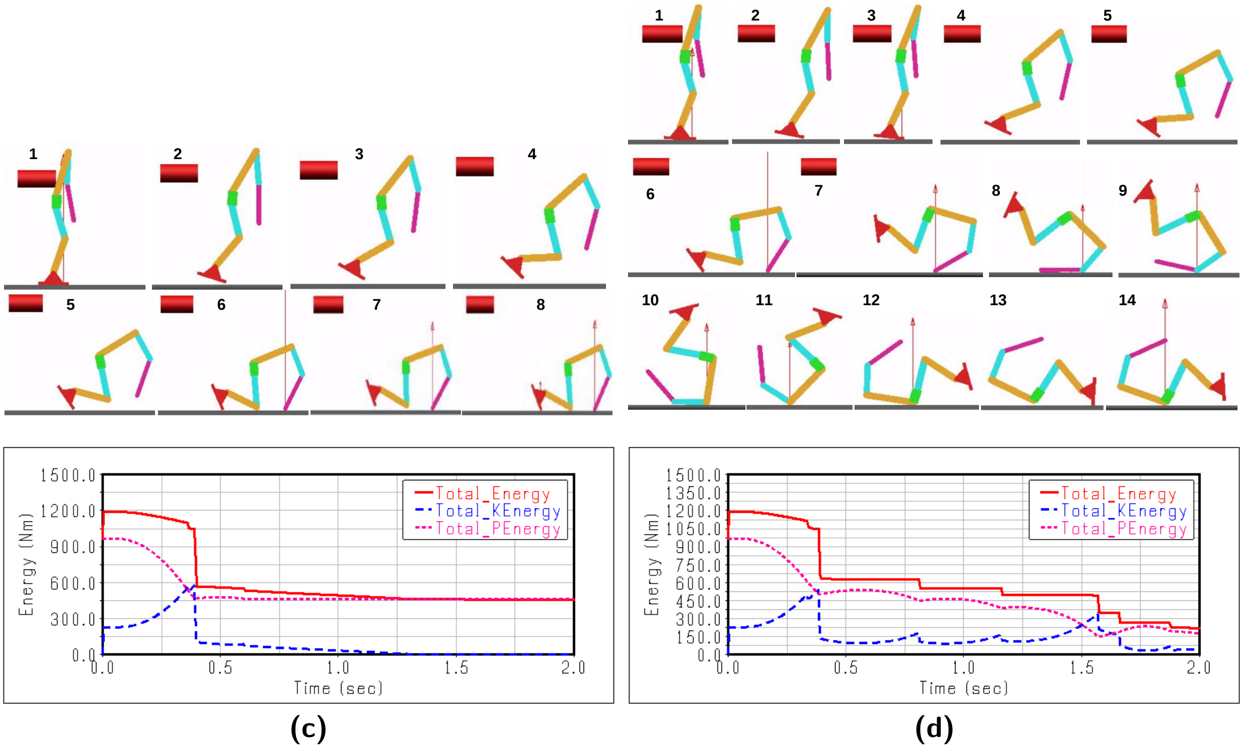
The simulation was carried out with MSC Adams 2013.2 installed on a laptop which has 4 cores, 8.0GB RAM, and 2.2GHz processor speed.

12-6-2 With and Without Energy Injection

Figure 12-12 shows the results of controlled fall simulations obtained with the newly proposed controller. In Fig. 12-12a, the snapshots at the top and the evolution of energy at the bottom plot denote the results obtained without injecting any additional energy (nEJ) into the system, during its fall. Similar results are shown for a controlled fall simulation with an additional injection of energy (wEJ) in Fig. 12-12b. From the snapshots, we can see that the lack of sufficient energy prevents the nEJ system from successfully completing its work which involves rolling over the secondary contact (end of lower arm), and hence it is not able to roll over successive contacts as planned. While with wEJ, due to the additional injection of energy, the system is able to overcome the *energy drought* condition. As a result, it exhibits successful rolling over the potential contacts as planned, generating a rolling motion. The energy evolution plots also suggest the same with nEJ. One major energy dissipation (ΔE = 450 J) is observed during the knee and hand contacts with the ground, following which no significant dissipation is observed. Whereas with wEJ, apart from the major energy dissipation (ΔE = 375 J) observed during the lower arm contact with the ground, subsequent energy dissipations due to the elbow (ΔE = 100 J), shoulder (ΔE = 75 J), backpack (ΔE = 150 J), and foot contacts (ΔE = 75 J) are seen, resulting in much higher dissipation of energy (total) and in a gradual manner. This ensures more energy being distributed over multiple contacts which in turn reduces the impact forces during fall.

12-6-3 Controller Tracking Performance

The optimal parameters for efficient rolling computed using the ORC in Section 12-3 are tracked by setting appropriate targets for Task 1 and Task 2. Once the fall is predicted to be inevitable, the controlled fall is initiated with the whole body control scheme tracking the optimal parameters until the lower arm is very close to the ground. This value is taken to be 0.05 m for the simulations carried out in this paper. The tracking actions are suspended once the hand reaches the set threshold, and the robot is let to fall over on its natural dynamics with the joint controller holding the robot's current posture. Figure 12-13 shows the controller's tracking of the hand's x and z position (first row) and the tracking of θ_E and θ_F (bottom row). In the above plots, we can observe smooth tracking of the hand's position to its optimal value and minimal error at the cut-off instant. Similarly, in



(c) (d)

Figure 12-12: Adams simulation results of SPR model: (a) without energy injection, and (b) with energy injection, where the first row shows the snapshots and the second row shows the potential, kinetic and total energy plots during the course of controlled fall.

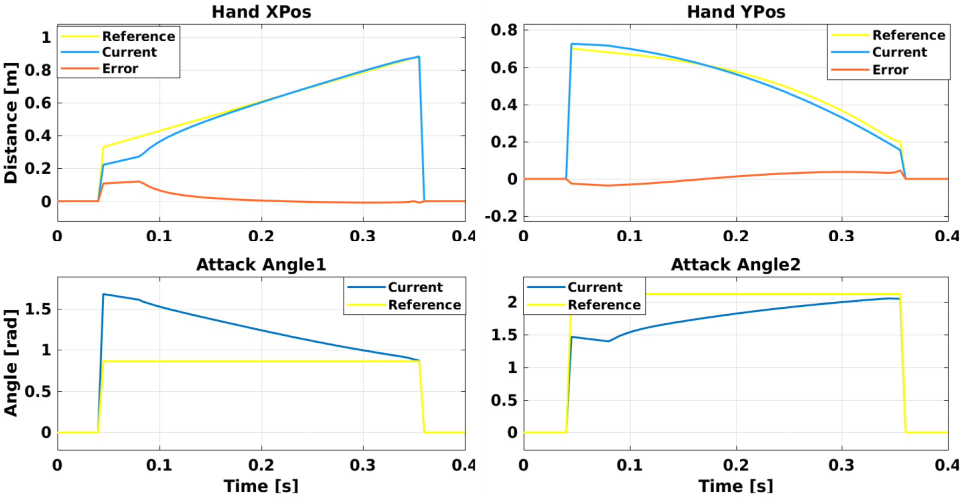


Figure 12-13: Optimal parameters tracking during the controlled fall (with energy injection).

the plots c and d, the optimal attack angles are reached smoothly with less error at the instant of cut-off.

12-7 Application to Humanoids

In this section, the proposed online rolling motion generator is applied to the WALK-MAN humanoid model for two reasons. First, to evaluate the effects of applying the proposed controller on a falling over humanoid. Second, to make a quantitative performance comparison between the new fall control scheme and our previously proposed controller. The simulation settings are maintained the same as in [69] to make an effective comparison of the results.

12-7-1 Simulation Scenarios

Three different scenarios are considered in total to ascertain the effectiveness of the proposed controller as follows: 1) simulation carried out with our previous best controller, which is a combination of ES and energy distribution (ED), 2) controlled fall simulated with the newly proposed controller (EIS + RED), a combination of ES and energy distribution with EDPH, without adding any external energy and 3) proposed EIS + RED type control with additional energy injected into the system. Since the first scenario had already been analyzed extensively in our previous work, only the remaining two scenarios are considered for evaluation. The results obtained are compared between scenario 2 and 3 to understand the effect of injecting energy to a falling humanoid, to verify if the results obtained are similar to those of the SPR model shown in the previous section.

Also, the results are compared with scenario 1 to evaluate our hypothesis, i.e., rolling over multiple contacts reduces the impact forces significantly. The results are presented and discussed in detail in the following subsections.

12-7-2 Energy Comparison

The robot is subjected to a disturbance of 1200 N, applied for 50 ms at 0.9 m height from behind to make it fall over, and the proposed controller is activated upon fall prediction.

In Fig. 12-14, the snapshots showed are taken during the controlled rollover simulation of a forward fall, with energy injection. The snapshots display the robot rolling over its left arm shoulder, and this is due to the fall direction which was predicted to be $\approx 5°$ to the robot's left side. Snapshots 2-5 show the humanoid crouching and modifying its left arm pose, resulting in a whole body motion of the humanoid to facilitate it. The crouching action is due to ES, and the latter pose modification is because of the control actions generated by RED. In 6, we can observe the final configuration of the left arm, where its hand (E) is just making contact with the ground, followed by the wrist and elbow (F) contact in 7, shoulder (G) contact in 9, and finally, in 10-12 the robot rolls over its head and backpack protection frame, to its left side. The complete controlled fall actions generate a forward left side shoulder rolling over motion. The primary contacts, i.e., knee (C and D) were

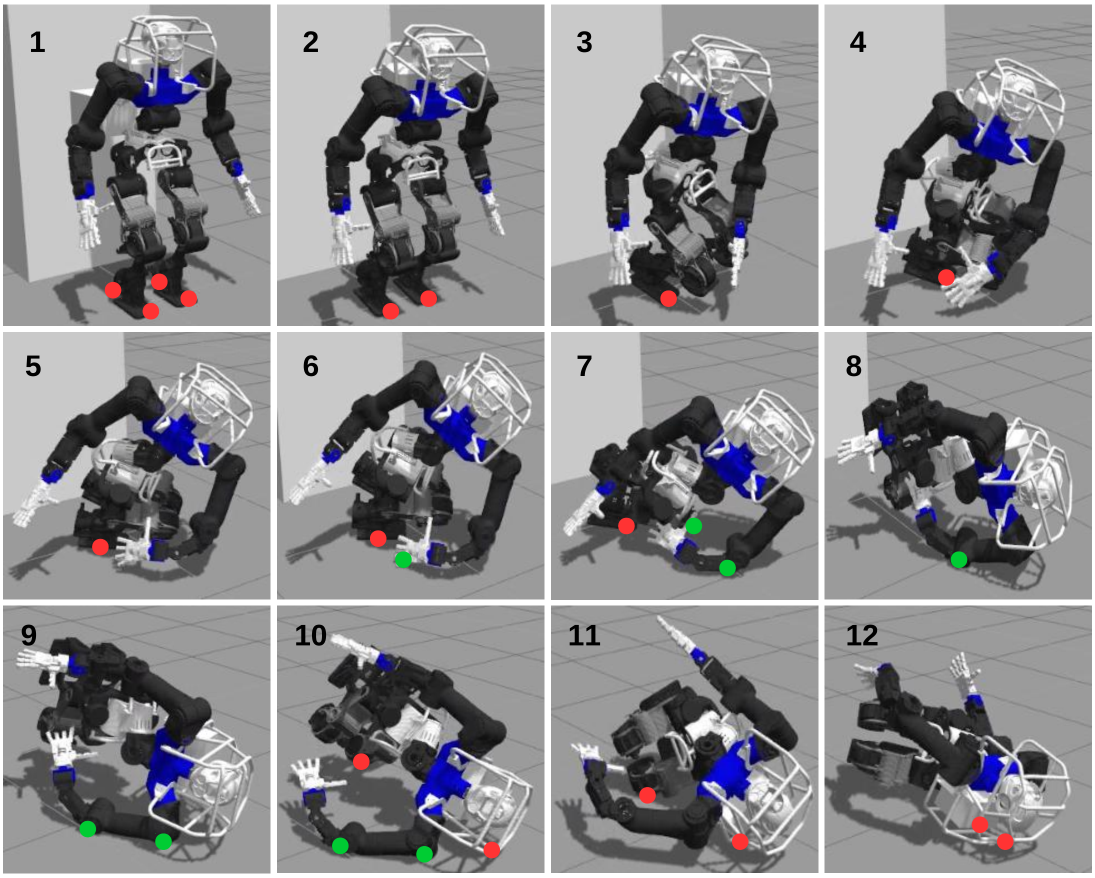


Figure 12-14: Simulation snap shots taken during the left side shoulder roll over of the robot with addition injection of energy into the system. The green and red dots denote the planned and unplanned contacts respectively.

observed only in 7 and were found to be closer to the ground in 5-6. In order to avoid an energy drought situation during this motion, energy is injected into the system using e_v obtained from the ORC.

The simulation snapshots shown in Fig. 12-15 demonstrates the controlled fall over motion of the humanoid without any additional injection of energy to the system. Comparing Fig. 12-14 and Fig. 12-15, we can see that in the latter one of the planned contact, i.e., the shoulder contact is not made during the falling over motion. Immediately after the hand and elbow contacts, the humanoid falls over to its left side due to the lack of momentum in the falling direction. The aforementioned observations are due to the lack of sufficient energy in the falling direction of the system, which is not the case with scenario 3. However, unlike the SPR case where the model's rollover motion was restricted in the planar direction, here the humanoid being a 3-dimensional (3D) model rolls to its side due to the CoG shift both in scenario 2 and 3, except that in scenario 3 the humanoid completely rolls over all planned contacts due to the additional energy.

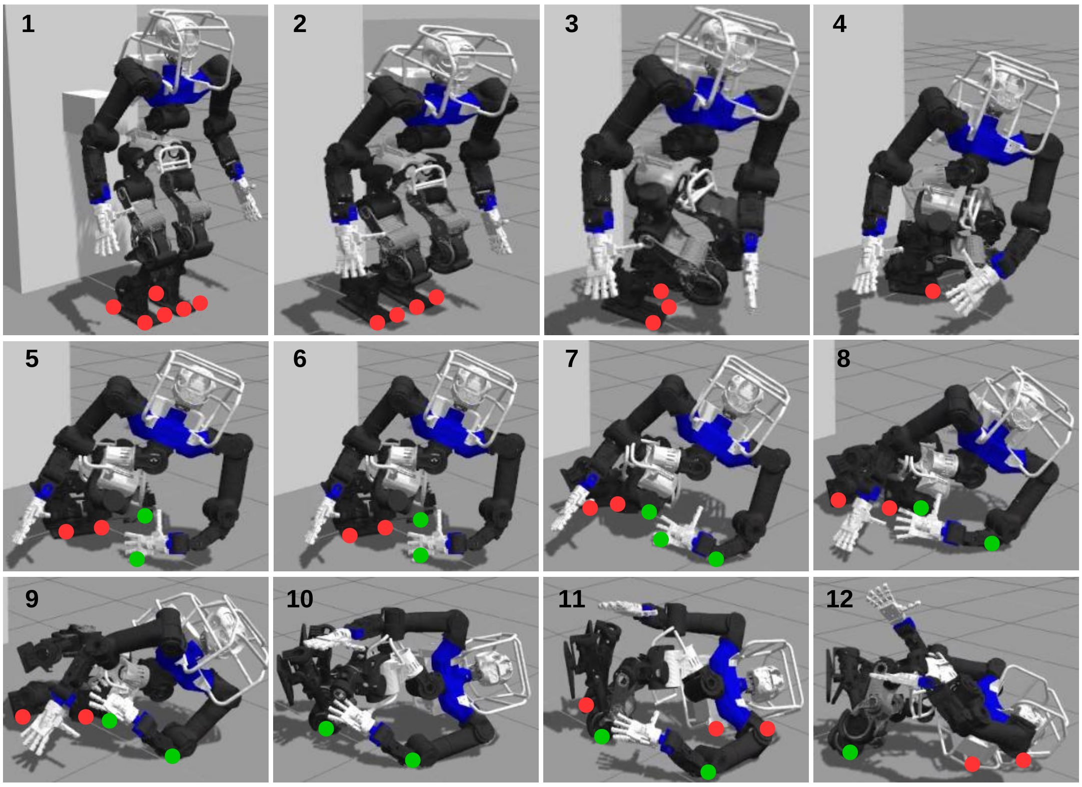
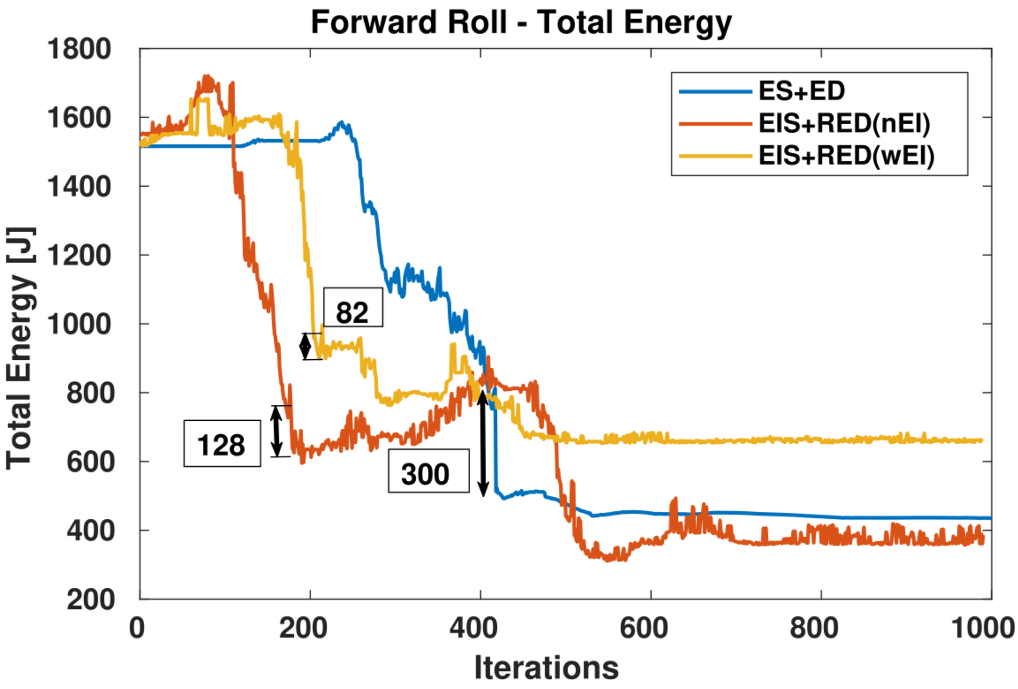
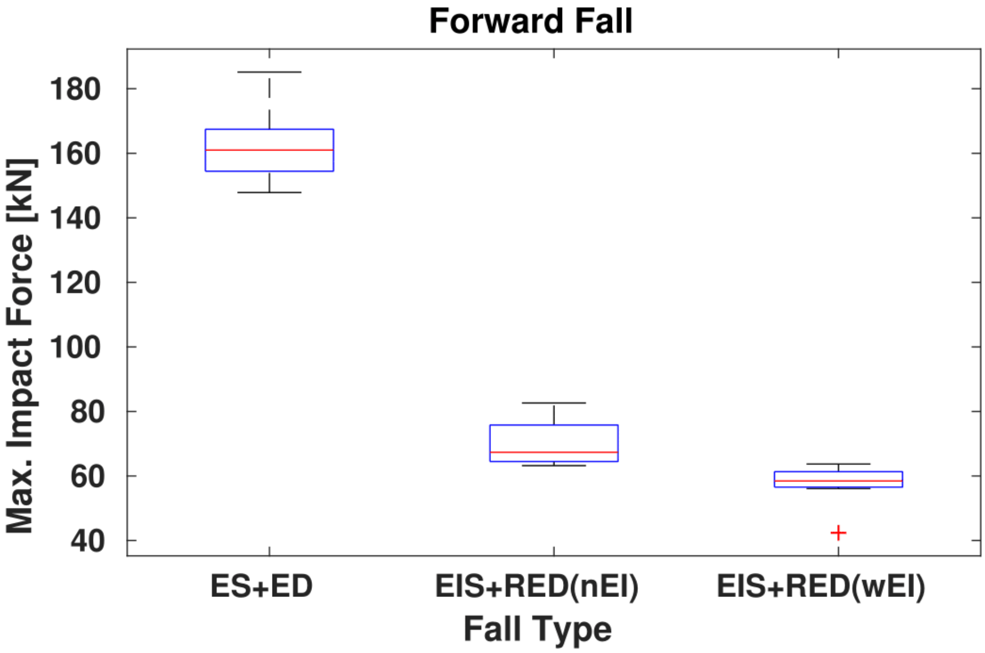


Figure 12-15: Simulation snap shots taken during the left side shoulder roll over of the robot without any additional injection of energy. The green and red dots denote the planned and unplanned contacts respectively.

The plots in Fig. 12-16a compares the evolution of energy between three types of controlled fall actions. Unlike our previous method (ES+ED), we can clearly see three to four energy dissipation points between 200-500 time steps in the total energy plots of EIS+RED(wEI) and EIS+RED(nEI) respectively. This denotes the successive contacts made by the robot and its corresponding dissipation of energy. The maximum amount of energy dissipated was found to be relatively less for EIS+RED(wEI) (82 J) when compared to EIS+RED(nEI) (128 J). This indicates that though e_v increases the total kinetic energy, due to rollover motion the number of contacts is increased which in turn reduces the energy dissipated per contact. For both the conditions, high energy dissipation is observed in the beginning when hand contact was made with the ground. Relatively, higher dissipation of energy (300 J) was observed with ES+ED at the end, due to the static nature of hand contacts. This shows clearly that the proposed method dissipates 72.6% more energy during the forward fall over.



(a)



(b)

Figure 12-16: Comparison between ES+ED, EIS+RED(nEI), and EIS+RED(wEI) type falls: (a) Energy dissipation and (b) Statistical plot of the impact force obtained over 10 trials.

12-7-3 Impact Force Comparison

Since the impact force can vary due to the robot's joint compliance, disturbance, simulator (physics engine, contact model), etc., 10 data sets are collected for the forward fall direction, and the results are compared between different controlled fall actions as shown in Fig. 12-16b. Overall, we can observe that the impact force (F_{imp}) has been reduced considerably with EIS+RED(wEI) and EIS+RED(nEI) control when compared to the ES+ED type control. Less difference is observed between EIS+RED(wEI) and EIS+RED(nEI) with respect to the magnitude of F_{imp} and this could be due to the fact that in both the cases the humanoid rolls over its side, except for the fact that in the former case an additional contact is made with the robot's shoulder. Also, the variance of F_{imp} in the former case is relatively lower when compared to the latter, suggesting that with EIS+RED(wEI) more consistent controlled fall actions can be taken. With the proposed controller 63.7% of reduction in F_{imp} for the forward fall was observed when comparing ES+ED to EIS+RED(wEI).

12-8 Other Direction Rolls

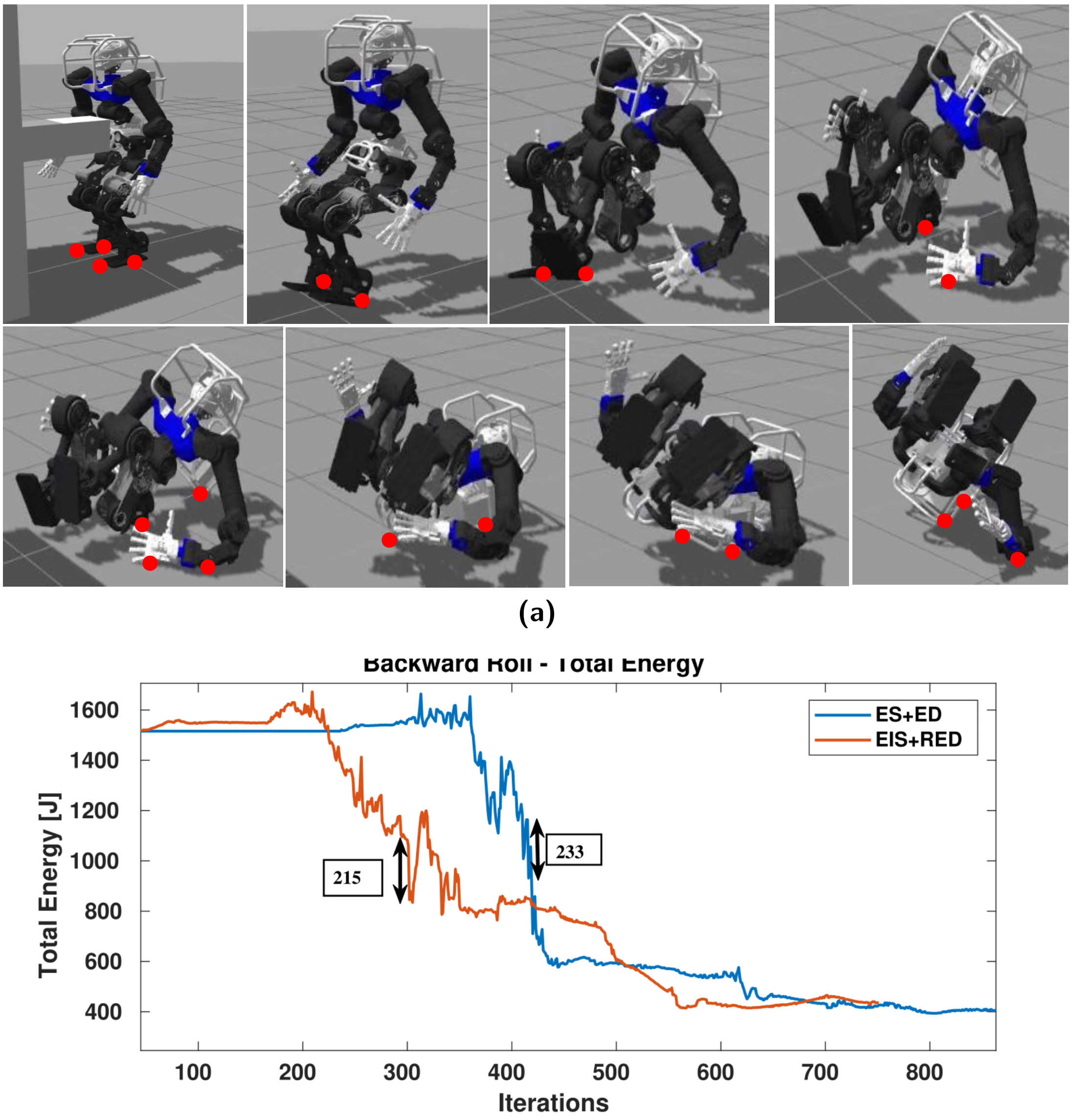
12-8-1 Backward Fall

The simulation snap shots for backward roll over obtained with EIS+RED controller are shown in Fig. 12-17a. Though the roll over can be done with either left or right hand during the backward fall over, here left hand has been considered. Snapshot 4 shows the robot making primary contact with the left hip and hand. The former contact is due to the squatting motion generated by means of energy shaping and the latter is due to the distribution polyhedron. Following the above contacts, wrist, elbow, and protector frame contacts are made in 5, 6, 7, and 8 respectively. Due to the protrusion of humanoid's protector frame and the joint limitation of waist roll, no shoulder contact of the left hand is observed. Due to the convexity created with H and I contacts discussed in Section 12-4-1, the humanoid continues to dissipate the remaining energy by rolling over them.

The evolution of energy during the backward fall over of humanoids is shown in Fig. 12-17b and it is compared to those obtained with ES+ED type controller. In the case of ES+ED, high dissipation of energy (233 J) is observed during the hand contact, whereas with EIS+RED, the value is slightly less (215 J) and it is dissipated across the hip contact which is relatively stronger to take high impacts. Also, the peak around 400th time step with EIS+RED shows that the humanoid continues to roll over after the first contact.

12-8-2 Side Fall

Figure 12-18a shows the snapshot taken during the right side roll over of humanoids. Snapshots 2 and 3 show the crouching motion and the movement of right arm due to the



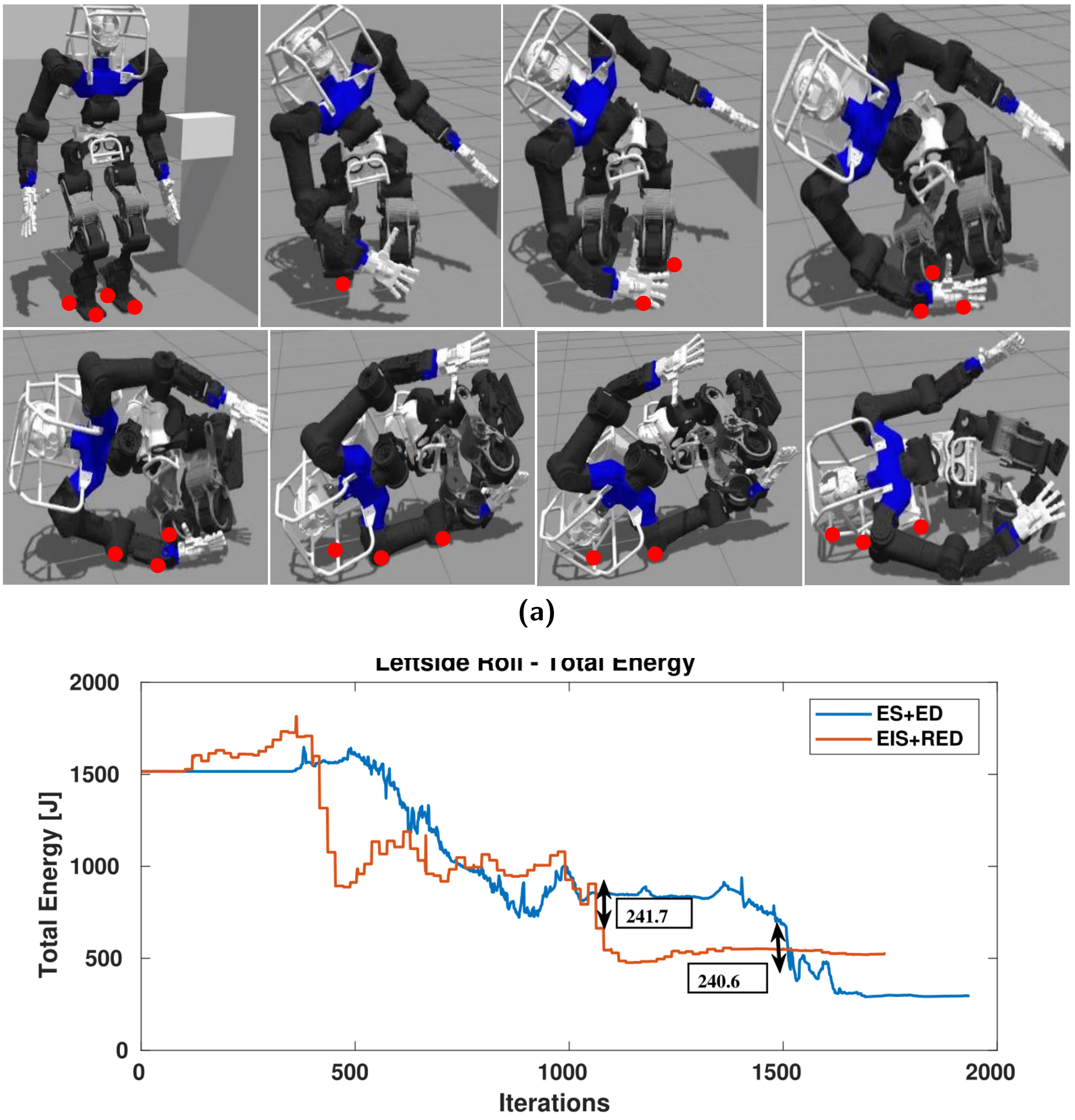
(a)

(b)

Figure 12-17: Simulation results of backward fall are shown here as follows: (a) simulation snapshots taken during the controlled fall and (b) energy plots comparing the evolution of energy between the controllers ES+ED and EIS+RED.

energy shaping and distribution techniques, resulting in the right knee and hand contacts. This is followed by the wrist, elbow, and shoulder contacts in 5, 6, and 7 due to the energy distribution polyhedron. Finally, the humanoid makes contact with the protector frame during the course of its roll over.

No significant difference is observed in the maximum value of energy dissipated between ES+ED and EIS+RED type fall overs. This could be due to the fact that with ES+ED, even if the contacts are not planned to roll over passively multiple contacts are made after the secondary contact with hand. However, in the case of EIS+RED, multiple contacts are planned optimally and controlled actively to roll over, yielding more consistent results when compared to ES+ED.



(a)

(b)

Figure 12-18: Simulation results of left-side fall are shown here as follows: (a) simulation snapshots taken during the controlled fall and (b) energy plots comparing the evolution of energy between the controllers ES+ED and EIS+RED.

Chapter 13

Discussion

As discussed in Chapter 1, fall over is one of the major issues to be addressed, in order to have humanoids operable either independently or alongside humans, in human-made environments. In reference to the above problem, we first proposed an online energy based falling over control for humanoids to mitigate the damages in Chapter 11. The proposed concept uses energy shaping and distribution techniques which are inspired by one of the well known break-fall motion Ukemi. Notable features of the controller were instantaneous motion planning for multiple directions with a single model and adapting it according to the disturbances and the resulting robot's dynamic motion. The controller's ability to reduce the energy dissipation and thereby mitigate the impact forces were successfully verified with the humanoid WALK-MAN by subjecting it to disturbances in both sagittal and frontal planes. Though the controller reduces the impact forces considerably in reference to free fall, the motion is relatively static, and the robot makes less number of body/ground contacts when compared to that of Ukemi. Exhibiting multiple contacts during the falling over motion could reduce the energy dissipated per contacts and this, in turn, can reduce the impact forces further as observed in break-fall motions like Ukemi.

To verify the above notion, a new controller was proposed in Chapter 12, to make successive contacts, to generate a rolling like motion. The conceived hypothesis is also inspired by the human rolling motion observed during Ukemi. Considering the potential contacts to roll over as knee (primary contact), hand (secondary contact), elbow, and shoulder, an online rolling controller (ORC), is proposed to compute the optimal contact position of the hand, the attack angles, and the additional energy to be injected (e_v). The optimal position and the angles are used to construct an energy distribution polyhedron, and e_v is considered during energy shaping. The proposed controller with the optimal parameters (EIS+RED) was first verified with the segmented planar robot (SPR) model, and then it was applied to the WALK-MAN humanoid.

Control tracking limit: With SPR model, the proposed controller showed good tracking performance both in following the desired contact position and the attack angles. Also, a significant difference was observed between without EIS+RED(nEI) and with energy injection EIS+RED(wEI). The former case resulted in less number of contacts, due to the lack of sufficient energy to overcome the work required in rolling over the hand contact. While the latter, due to the injection of additional energy into the system, makes multiple contacts and exhibits a rolling like motion. The evolution of energy also displayed multiple dissipations of energy over different contacts, unlike the EIS+RED(nEI) case.

When the proposed controller was applied to WALK-MAN, the tracking error was relatively high when compared to its performance with the SPR model. This is attributed to the combined limitations of the PID control and the actuator power towards high-speed tracking during the fall over motion. In Appendix A-3, the effect different PID settings and actuator power levels have on the controller's tracking performance is presented briefly. Between EIS+RED(nEI) and EIS+RED(wEI), a minimal difference was observed in the energy plot and the statistical plot of the impact force plot. However, the former had a high variance suggesting the later to be resulting in more consistent controlled fall motion. The minimal difference is also due to the fact that WALK-MAN's motion is not restricted to planar, unlike SPR, and hence it rolls over to its left side even if it doesn't have enough energy to roll over the shoulder contact. This results in dissipating some amount of energy over the protection frame contacts and thereby reduces the impact force. To summarize, both EIS+RED(nEI) and EIS+RED(wEI) type controls reduce the impact forces significantly when compared to ES+ED, with EIS+RED(wEI) showing more consistent rolling like motion.

Limitations and Open questions: The controllers proposed in this work have been developed to target some of the major requirements of humanoid fall over controllers. However, there are few shortcomings with the proposed controllers which needs to be addressed and some open issues which requires careful thought. Some of the aforementioned ones are discussed succinctly below.

- *Proposed control strategies limits*: From the controller performance analysis made in Section A-2 we can understand that the tracking performance is limited by PID gains and actuator power. Another limitation which is related to the aforementioned ones is the magnitude of impulses which the different energy-based control strategies can handle. This can define the boundary for each proposed strategies and may even invoke the synthesis of new favorable combinatorial according to the type of disturbances. For instance, in the case of high impulse disturbances, it is impossible to minimize the energy using the shaping strategy since the crouching action of it is severely limited by the actuator power. In such scenarios, it could be beneficial to make a step towards the falling direction and execute the energy distribution strategy which is relatively quick and effective as observed in Ukemi. Similarly, in the case of low impulses probably it is better to use the ES+ED strategy instead of EIS+RED since the latter may infuse more energy into the system and this could affect the body parts. In the above situation, using ES+ED is better only if the robot's body parts can take the impact force without any major damage. To precisely determine the boundary of each strategy and the new favorable combinations it is necessary to evaluate the effectiveness of each strategy by subjecting it to different magnitudes of impulses.

- *Controlled fall during dynamic motion*: Another important issue which has not been given enough consideration here is the ability of the proposed controllers to handle fall overs of humanoids from arbitrary configurations or during dynamic motions. In both the cases, the challenges are more about planning the motion online avoiding self-collisions, computation of optimal contact positions, and tracking them. In addition, in the dynamic cases, smooth transition from one controller to another one could also be crucial for controlled fall overs. Since the dynamic state of the robot also has to be considered at the instantiation of controlled fall.

- *Quantification of damages*: Finally, methods to quantify the damage each humanoid can sustain immediately after their fall overs has to be developed to have a realistic evaluation of the proposed controllers. For instance, a certain magnitude of impact force can be safe to one body part, but it could be devastating to another part of the humanoid. In such cases, even though the controller reduces the impact forces, it still doesn't achieve its primary goal of keeping the humanoid in operable condition even after a fall over. Both in the newly proposed and the existing controllers, damage quantification has never been done before. Such methods can be used both to evaluate the controller realistically and to consider them actively inside the controllers to generate optimal control actions.

Chapter 14

Concluding Remarks and Future Work

14-1 Concluding Remarks

In the presented work, we have tried to address the problems associated with the fall recovery of humanoids. Dividing the above issue further into three sub-issues such as fall prediction, controlled fall, and fall recovery, the first two sub-issues has been primarily focused here.

In relation to fall prediction, a novel technique has been proposed here to combine the conventional sensors (inertial measurement unit (IMU) and encoders) with the foot pressure sensor (FPS) and vision sensor, to predict humanoids fall over in a reliable and agile manner under different conditions. The proposed prediction principle is inspired by how humans use the balance sensory systems to monitor their state with respect to the environment and thereby make optimal decisions. The advantages of redundant sensory systems observed in humans such as reliability, agility, generic nature, and optimality have been exploited by combining the different sensor features using Kalman filter. With extensive numerical analysis, we demonstrated the aforementioned uses of redundant sensors and their varied performances on different terrains and also with simple dynamic cases. It was also shown that there is no single best feature for fall over prediction, and an optimal performance in generic scenarios can be obtained by fusing the multiple features with Kalman filter-based (KF). Apart from that, the proposed fall prediction method was found to be robust in handling disturbances from diverse directions while the robot was standing on different terrains, and also against sensing noise. The above summary of performance was successfully verified through real experiments conducted with the WALK-MAN robot over flat and uneven terrains.

With a reliable, versatile, robust, and agile fall prediction technique, the focus of the research work was shifted towards dealing with the controlled fall of humanoids. In reference to that, energy-based controllers were proposed in this work to mitigate the impact forces which humanoids sustain during their inevitable fall overs. Two controllers were proposed in total, and both are inspired by the break-fall motions, in particular, the Ukemi motion. From the above inspiration, three strategies based on energy are introduced here, and they are used in different combinations forming two novel controllers. The first one includes the energy shaping (ES) strategy to minimize the system's energy, and later, the distribution (ED) technique to distribute the minimized energy over multiple contacts exploiting energy distribution polygon (EDP). In addition, an orientation control of the hands effectively protects the fragile wrist joints during the fall. The proposed fall control is comparatively evaluated by two controllers, ES and ES+ED, of which the proposed ES+ED method was found to be significantly better.

Though the above controller reduced the impact forces considerably, the resulting motion was rather static when compared to those of break-fall motions, and this is due to the less number of body/ground contacts made during the falling over motion. To generate a dynamic multi-contact motion like rolling, a novel controller was proposed with an additional strategy called energy injection. It has three major contributions: an online rolling controller to compute the critical rolling parameters and energy injection values, and the notion of an energy distribution polyhedron to realize the motion. The proposed controller was first evaluated with the segmented planar robot (SPR) model, in particular, to validate the significance of injecting energy (e_v) to a falling system. The results showed that with e_v it is possible to avoid energy drought and reduce the impact forces considerably by means of rolling. The controller was then applied to a full humanoid, and it is verified that the proposed controller can considerably reduce the impact force by generating rolling motion in comparison to the control without it (ES+ED). The newly proposed one (EIS+RED) exhibited a multi-contact rolling motion, and this reduced the maximum dissipation of energy per contact, which in turn reduced the impact forces significantly.

14-2 Future Work

14-2-1 Fall Prediction

The fall prediction method proposed in this thesis work in Part II has been verified to be generic in handling various disturbances applied from different directions, agile, robust to noises, reliable, and extensible to dynamic cases. However, as mentioned in Chapter 8, there are few limitations like explicit extraction of thresholds, cost functions to devise non-linear boundaries, and direct applicability to more complex dynamic motions like walking. Addressing the aforementioned limitations could be the future direction to make the prediction method more complete.

One possible option to alleviate some of the above limitations is the use of machine learning techniques, in particular, Gaussian process (GP). Since with GP, a variety of kernels can be formulated according to the characteristic of a particular feature, explicit extraction of thresholds and cost functions to define non-linear boundaries can be avoided. Also, in uncertain situations, reliable decisions can be taken with probability-based predictions. In addition, it is possible to model the noise associated with each sensor and consider it during the process of learning to make robust online predictions. Another important feature of GP is its online learning/adaptation of the model which could be useful in evolving the prediction method based on its online performances. Though GP seems to be promising in resolving the limitations it has to be developed and applied for fall over predictions to actively compare with the proposed method.

To predict fall overs during complex dynamic motions like walking, the existing features FPS, IMU, and optical flow (OPF) can be used directly by including the data related to walking during the training phase. This is possible since the evolution of the above features is relatively less direction specific when compared to those of center of gravity (CoG) and zero moment point (ZMP). Hence, the latter cannot be applied directly, which calls for either a new set of features which are less direction specific or a combination of different ones. This needs to be analyzed in order to consider them in the prediction method. Also, generalizing the method in predicting fall overs during different walking speeds is also a concern which needs to be addressed.

14-2-2 Controlled Fall

In the last part of this work, we introduced several energy concepts and actively used them to propose two different controllers to mitigate the damages incurred by the humanoids during their fall overs. Both the controllers were able to generate control actions, adapt them according to the robot's state and apply them online. Further, they were generic enough to handle both frontal and sagittal fall overs with less number of simplified models, making them easier to use. However, there are few missing things or certain characteristics of the controller which needs to analyzed more in detail and these have been touched upon in Chapter 13. Some of them could be potential research topics to pursue in the future to make the method more generic and reliable. Few of them are discussed below.

First, the limits of different control strategies proposed for controlled fall which hasn't been analyzed extensively in this work, mainly with respect to the magnitude of disturbance and the location on the robot where it is imparted. Since the performance of different strategies varies considerably over the disturbances and it is also heavily limited by the actuator power. Analyzing this can help us in defining the boundary clearly for each strategy and can even result in new combinations of different strategies which can increase the disturbance limit.

Second interesting topic to be considered is how to carry out controlled fall during dynamic motions. Apart from external disturbance induced fall overs, a majority of them occur during dynamic motions, and they are mostly induced by the manipulation or locomotion controllers. In such scenarios, it is even difficult to carry out controlled fall motion since the initial state of the robot can be any arbitrary configuration which is not the case reported here. Also, to safely fall over self-collision free actions have to be planned and generated online making the whole process more difficult. This requires considerable attention in executing the controlled fall actions during dynamic scenarios.

Third, is the planning and execution of a reactive controlled fall in an environment filled with different objects. In this case, the objective here could be to prevent damages to both the robot and its surroundings, i.e., to make use of the surrounding environment to stop the robot from falling over and stabilize its posture. To achieve them it is necessary to plan the contacts on different objects not just to minimize the impact force but also to arrest the fall over motion and to stabilize the robot. Fourth could be to develop a method to analyze and quantify any damages sustained by a humanoid during its fall over. This could be useful in evaluating efficiently any controlled fall strategy and also to make an effective comparison between different existing strategies.

14-2-3 Other Works

Final direction of the presented research work would be to develop a recovery controller which can plan a humanoid's whole body motion to recover from its fall over. The generated actions should be adaptable according to the initial configuration of the humanoid, should consider the joint, velocity, and torque limits, and should be collision-free. The ultimate goal of this research is to combine fall prediction, controlled fall, and recovery controller, and ensure that they work in tandem to realize successfully the prediction of an inevitable fall over of a humanoid, execution of controlled actions to minimize the damage, and finally recover from it.

Appendix A

Appendix

A-1 Versatility of SPR Model

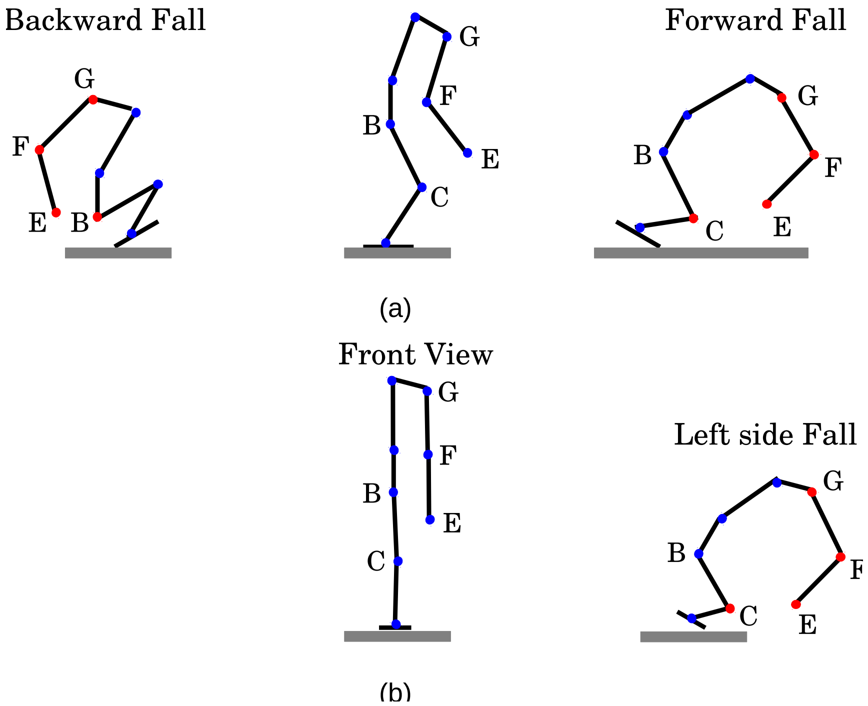


Figure A-1: The versatility of SPR model in planning the contacts for humanoids falling over in different directions is shown here. The planned contacts (red dots) are shown for different fall over directions: (a) sagittal falls which includes forward and backward fall overs, and (b) frontal falls which includes left and right side fall overs.

The versatility of the SPR model introduced in Section 2-3-3 is shown here in Fig. A-1. In the figure, C, E, F, G, and B represents the knee, hand, elbow, shoulder, and base contact in the SPR model. Figure A-1 shows the contacts considered for planning the rolling configuration for sagittal falls which includes forward and backward fall overs. In the case of frontal falls such as a left side fall over shown in Fig. A-1b, it is still possible to plan optimal rolling configuration for a falling over humanoid. The primary contact with C, followed by the secondary one with E can still be achieved in the frontal case due to energy shaping. Though SPR is a planar model it is possible to plan the contacts for different fall over directions and together with energy distribution polyhedron the model can be used to plan optimal configurations for omnidirectional fall over of humanoids.

A-2 Controller Performance Analysis

The performance of the controller in terms of its ability to move the selected contacts to the desired reference position is analyzed in this subsection. Figure A-2 shows the time evolution of the set reference, the instantaneous position and the tracking error of contact E in the first row. The plots only show the evolution of the variables during the tracking period, which is from 14.4s to 14.8s. As seen in the plots, the controlled motion is initiated at 14.4s and lasts until ≈14.8s, after which the controller is turned off to allow the robot to roll over with its natural dynamics. The optimal reference position for contact E is set using the online rolling controller (ORC), as discussed in Section 12-5.

In the PosX and PosY plot, (top row) we can see a smooth reference trajectory (red line) connecting their respective initial value to their target positions. The spline trajectory is generated considering 0.35s as the control tracking period, from the instant of fall prediction. Similarly, for the z coordinate, a reference trajectory is generated, but it is adapted online according to the knee position, during the entire tracking period. For all three plots (PosX, PosY, and PosZ), we can observe each coordinate's instantaneous value (green line) closely following the reference trajectory until the end of tracking period (14.8s). An error (black line) of 0.05m, 0.09m, and 0.095m are respectively observed for the XYZ coordinates of the hand position.

The bottom row plot shows the optimal attack angles set as references (red line) and the instantaneous value of the angles tracking their respective targets during fall over. Similar to the hand position coordinates, a smooth spline trajectory connecting the initial value to their respective target value is generated. An error of 0.19 and 0.28 rad was observed for θ_E and θ_F respectively, at the tracking cutoff instant. The error shows the convergence of instantaneous values to their respective references during the controlled fall motion. However, the error is slightly high for the x coordinate at 15.5s, and this could be due to the fact that Task 2 being controlled under the nullspace of Task 1.

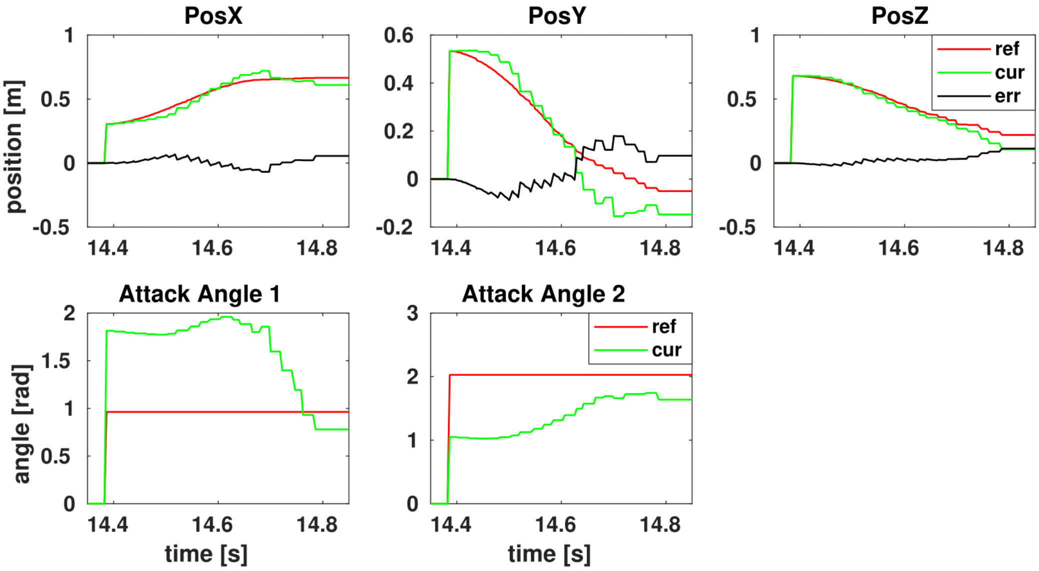


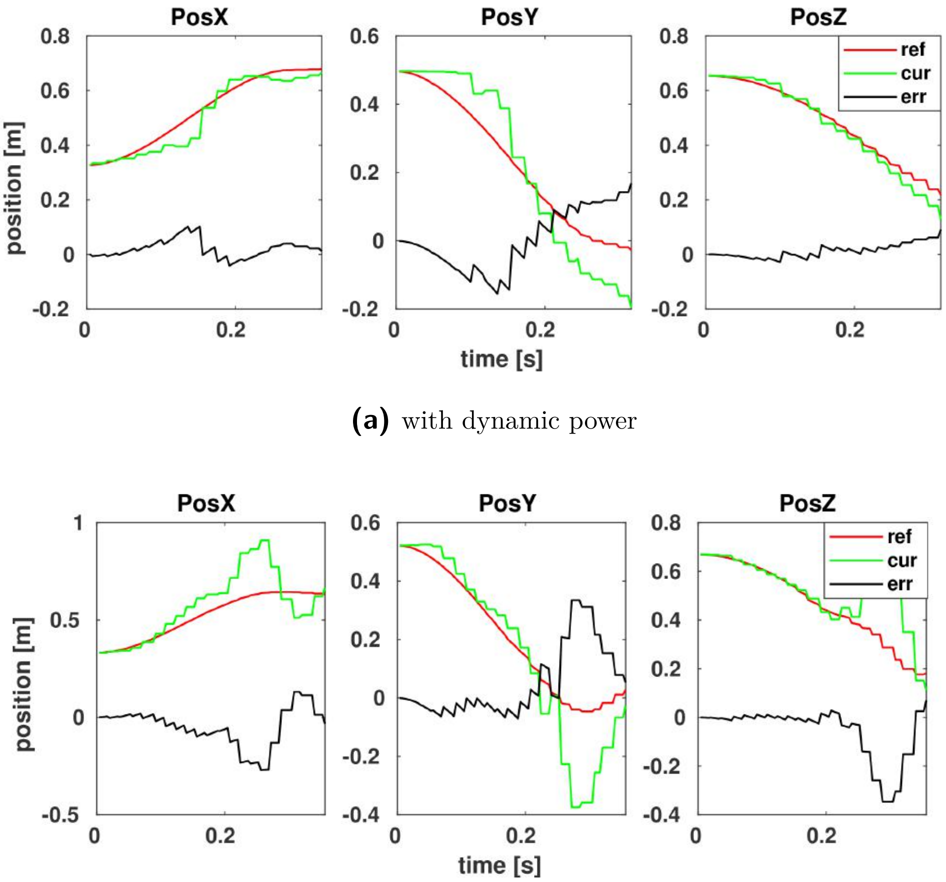
Figure A-2: Controller performance analysis: First row shows the optimal reference, the instantaneous position, and the tracking error of the left hand position, while the second row shows the tracking of the attack angles θ_1 and θ_2.

A-3 Dynamic vs Unlimited Power

The falling over of humanoids is incredibly quick demanding any set of actions to be planned and executed within a short interval of time. While the planning of optimal control actions can be done with smart techniques, high processing power, etc., the execution of agile actions is limited by the actuator power. Keeping the low level PID gains constant and executing the proposed EIS+RED controller with two different set of powers, peak and unlimited, certain differences were observed in the controller's performance.

A-3-1 Position Tracking Comparison

Figure A-3 compares the tracking performance of the optimal position of hand ($\mathbf{p}_{\mathrm{E}}$), one of the CRP computed using ORC, between dynamic and unlimited powers. The reference, current, and error of $\mathbf{p}_{\mathrm{E}}$ is denoted by the red, green, and black lines respectively. From fig. A-3a it can be seen that with dynamic power the error for $\mathbf{p}_{\mathrm{Ey}}$ is considerably higher (0.17m), followed by relatively lower error value (0.11m) for $\mathbf{p}_{\mathrm{Ez}}$. With unlimited power shown in fig. A-3b, a high error value of 0.07m is reported for $\mathbf{p}_{\mathrm{Ey}}$ at the instant of secondary contact. Though the error magnitude is less in the case of unlimited power, one can notice high deviations from the reference trajectory before making the secondary contact. This is probably due to the constant PID gains set for the joint, retuning them considering the unlimited power could help in avoiding such deviations.



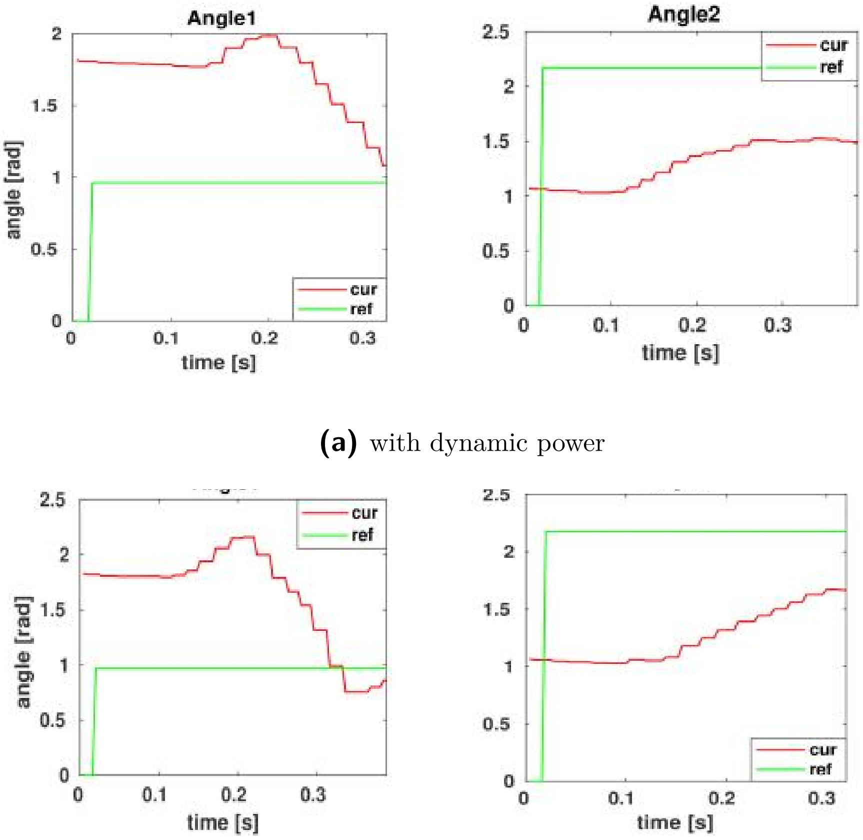
(a) with dynamic power

(b) with unlimited power

Figure A-3: The effect actuator power has on tracking the optimal position of hand ($\mathbf{p}_\mathrm{E}$) with the proposed controlled fall is shown between two different powers: (a) dynamic power, and (b) unlimited power.

A-3-2 Orientation Tracking Comparison

The tracking performance of the proposed EIS+RED controller in the case of attack angles is compared between the two types of powers in Fig. A-4. In the figure, angle 1 and 2 refers to θ_E and θ_E respectively. The observations made here are similar to those made earlier with the position tracking. With the unlimited power the tracking error for both the attack angles is relatively minimized just before the contact of E. However, the error is relatively high when compared to that of position tracking and this is due to the high priority given to the control of hand position.



(a) with dynamic power

(b) with unlimited power

Figure A-4: The effect actuator power has on tracking the attack angles (θ_E and θ_F) with the proposed controlled fall is shown between two different powers: (a) dynamic power, and (b) unlimited power.

A-4 Effect of Late Fall Prediction

Fall over of humanoids in general is instantaneous and incredibly quick limiting the time available to predict it, plan and execute control actions. In the case of human size humanoid robots, due to their high inertia and the bounded actuator power available to the joints, the speed at which any action can be executed is very limited. This calls for agile prediction methods such that enough time is available for the humanoids to perform any control actions in a swift and stable manner. The effect of late fall prediction is analyzed here by delaying the instantiation of a controlled fall strategy, here the crouching action of the humanoid has been considered, which is nothing but the lowering of CoG. Keeping the individual joint actuators at its maximum power, the crouching action of humanoids is initiated at various tilting angle during the course of its fall over as shown in Fig. In the figure, the x-axis denotes the angular position at which the crouching action was initiated, i.e., the fall was predicted to be inevitable, and the y-axis represents the total reduction in the height of CoG observed just before the humanoid hits the floor. The crouching action basically reduces the potential energy of the system by bringing down the CoG and this reduces the total energy of the system, which in turn mitigates the impact forces during the ground contact. Hence, the higher the reduction of CoG height the lower is the impact forces. In the figure, the blue dot represents the total reduction in CoG height obtained for starting the crouching action at certain fall over angle and the red line is the fitted curve to predict the trend. From the fitted curve we can see that as the crouching motion is initiated at a later angle, i.e., it is delayed, the total reduction in CoG height reduces and beyond 0.55 rad the reduction is less than 15% making the crouching action ineffective to mitigate the impact forces. Similar observation was reported by Robinovitch in [37], who conducted several trials with humans to evaluate the effectiveness of squatting motion in mitigating the impact. He showed through his results that the more humans delayed their squatting motion, the higher was their acceleration just before hitting the soft mat, which suggested that they will be subjected to higher impact and their action would be ineffective.

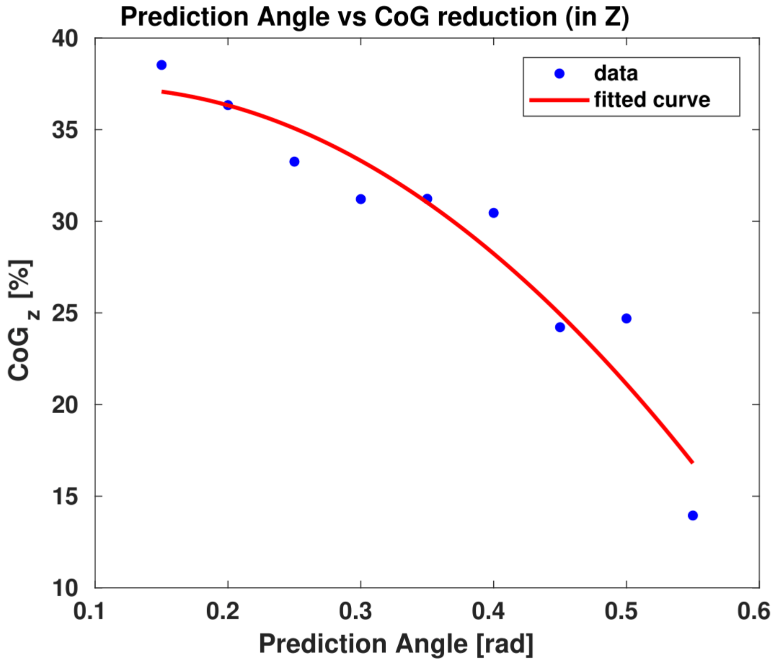
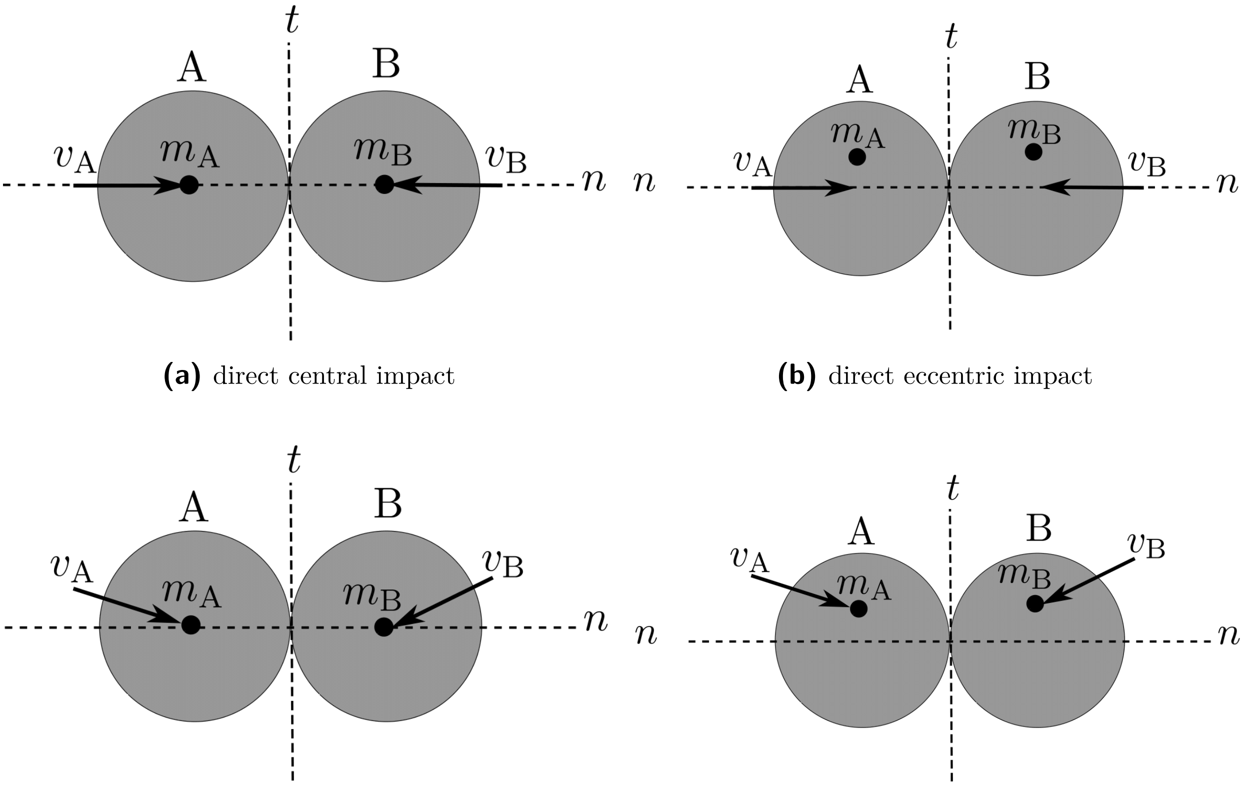


Figure A-5: The effect late fall prediction has on the efficiency of crouching motion, one of the controlled fall strategy, is shown here. The blue dots denote the total reduction in CoG height for initiating the crouching motion at certain fall prediction angle and the red line is the fitted curve to predict its trend.



(a) direct central impact

(b) direct eccentric impact

(c) oblique central impact

(d) oblique eccentric impact

Figure A-6: Different types of impact between sphere A and B are shown here, with n representing the line of contact and t the tangential plane perpendicular to n.

A-5 Impact and its Different Types

A collision between two objects occurring for a very short interval of time and during which relatively large amount of forces exerted between each other is called an impact. The common normal to the surfaces of both the colliding objects in contact during the impact is called the *line of impact.*

A-5-1 Impact Types

Based on Center of Mass Location

Depending upon the center of mass (CoM) location of each object with respect to the line of impact the impacts can be divided into two types: central and eccentric impact. If the CoM of both the objects are on the line of impact then it is called central impact and if the CoM is offset from the line of impact during collision then it is termed eccentric impact. For simplicity, the impacts has been considered to be of central impact type for the computation of critical rolling parameters in Section 12-3.

Based on Velocity Direction

In both central and eccentric impacts, the impact can be further divided into two depending on the velocity direction of the colliding objects. If the object's velocity is directed along the line of impact then it is called direct central impact. If the velocity of either one or both the objects is directed in a line different from that of the line of impact then it is called oblique central impact.

A-5-2 Impact Analysis

Since the presented work deals with the fall over of humanoids, its impact scenario can be represented succinctly by a point mass sphere coming in contact with the ground. As stated in Section 10-1, reducing the initial energy of the system reduces the dissipation of energy (δE) and this in turn lowers the impact forces.

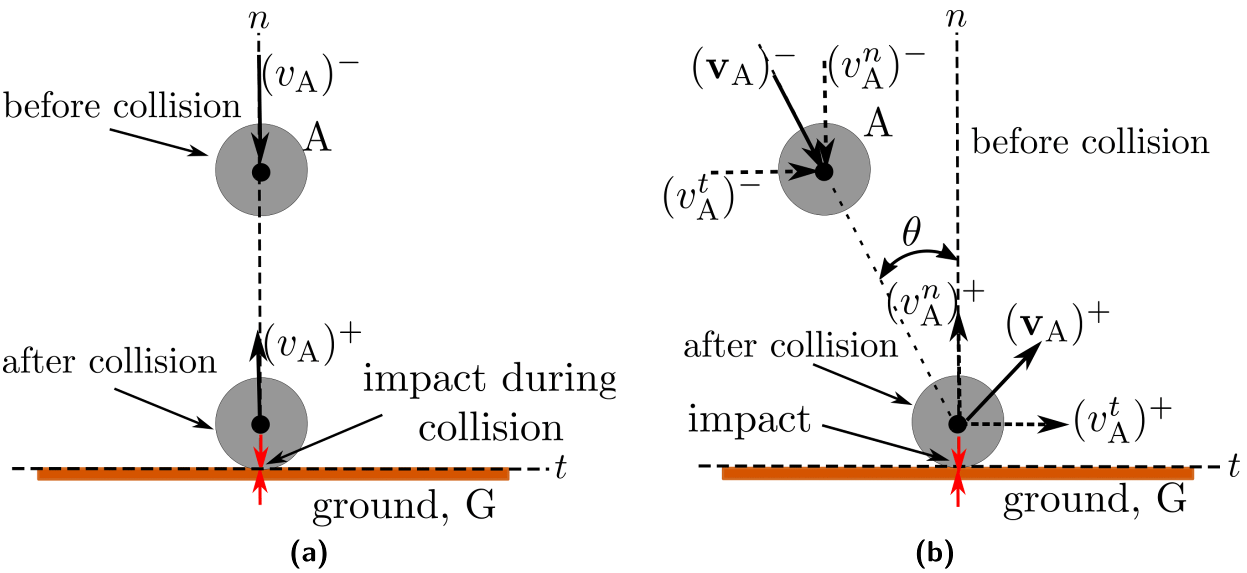


Figure A-7: The change in the velocity of sphere A with the ground G before and after collision is shown for two different types of impact: (a) direct central impact, and (b) oblique central impact. $(v_\mathrm{A})^-$ and $(v_\mathrm{A})^+$ represent the velocity of sphere A before and after the collision. The component of $\mathbf{v}_\mathrm{A}$ along the line of impact n is denoted by (v^n_A) and (v^t_A) is along the tangential plane t.

Direct Central Impact

In the case of direct central impacts shown in Fig. where v^-_A and v^+_A represent the velocity of sphere A before and after the ground impact. Since v^-_A is directed along the line of impact n, δE can be computed as follows:

$$\delta \mathrm{E} = \frac{1}{2} m (v_{\mathrm{A}}^{2})^{-} - (v_{\mathrm{A}}^{2})^{+}, \tag{A-1}$$

where $(v_{\mathrm{A}}^{2})^{+}$ can be computed using the coefficient of restitution, e which relates the velocities of colliding objects as shown below:

$$e = \frac{(v_G)^{+} - (v_A)^{+}}{(v_G)^{-} - (v_A)^{-}} \tag{A-2}$$

Since the ground velocity is zero both before and after the impact, substituting $(v_G)^{+} = (v_G)^{-} = 0$, $(v_A)^{+}$ can be computed as $e.(v_A)^{-}$. The value of e depends on the material of the colliding objects.

Oblique Central Impact

With oblique central impacts, the object velocity direction is not aligned with the line of impact (n) and hence its projection along the direction of n, i.e., $(v_A^n)^{+}$ and $(v_A^n)^{-}$ should be considered instead of $(v_A)^{+}$ and $(v_A)^{-}$ in equations (A-1) and (A-2). The tangential component of $\mathbf{v}_{\mathrm{A}}$, i.e., $(v_A^t)^{+}$ and $(v_A^t)^{-}$ are perpendicular to n and hence they do not contribute to the change of the object's momentum due to the impact. Assuming no friction along the tangential direction $(v_A^t)^{-}$ can be taken as equal to $(v_A^t)^{+}$.

Bibliography

[1] Jianbo Shi et al. Good features to track. In *Proc. of 1994 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 593–600.

[2] Jean-Yves Bouguet. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporation*, 5(1-10):4, 2001.

[3] Giuseppe F Rigano, Luca Muratore, Arturo Laurenzi, Enrico M Hoffman, and Nikos G Tsagarakis. A mixed real-time robot hardware abstraction layer (r-hal). *Encyclopedia with Semantic Computing and Robotic Intelligence*, page 1850010, 2018.

[4] Takemasa Arakawa and Toshio Fukuda. Natural motion generation of biped locomotion robot using hierarchical trajectory generation method consisting of ga, ep layers. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 1, pages 211–216. IEEE, 1997.

[5] Ambarish Goswami. Postural stability of biped robots and the foot-rotation indicator (fri) point. *The International Journal of Robotics Research*, 18(6):523–533, 1999.

[6] Atsuo Takanishi, Masami Ishida, Yoshiaki Yamazaki, and Ichiro Kato. The realization of dynamic walking by the biped walking robot wl-10 rd. *Journal of the Robotics Society of Japan*, 3(4):325–336, 1985.

[7] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. Springer, 2016.

[8] Ambarish Goswami and Vinutha Kallem. Rate of change of angular momentum and balance maintenance of biped robots. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 4, pages 3785–3790. IEEE, 2004.

[9] Jerry Pratt, John Carff, Sergey Drakunov, and Ambarish Goswami. Capture point: A step toward humanoid push recovery. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 200–207. IEEE, 2006.

[10] Pierre-Brice Wieber. On the stability of walking systems. In *Proceedings of the international workshop on humanoid and human friendly robotics*, 2002.

[11] Jerry E Pratt and Russ Tedrake. Velocity-based stability margins for fast bipedal walking. In *Fast Motions in Biomechanics and Robotics*, pages 299–324. Springer, 2006.

[12] John Rebula, Fabian Canas, Jerry Pratt, and Ambarish Goswami. Learning capture points for humanoid push recovery. In *Proc. of 2007 IEEE-RAS Int. Conf. on Humanoid Robots*, pages 65–72.

[13] X Xinjilefu, Siyuan Feng, and Christopher G Atkeson. Center of mass estimator for humanoids and its application in modelling error compensation, fall detection and prevention. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 67–73.

[14] Andrea Del Prete, Steve Tonneau, and Nicolas Mansard. Zero step capturability for legged robots in multicontact. *IEEE Transactions on Robotics*, 34(4):1021–1034, 2018.

[15] Zhibin Li, Chengxu Zhou, Juan Castano, Xin Wang, Francesca Negrello, Nikos G Tsagarakis, and Darwin G Caldwell. Fall prediction of legged robots based on energy state and its implication of balance augmentation: A study on the humanoid. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 5094–5100. IEEE, 2015.

[16] Sung-Hee Lee and Ambarish Goswami. Fall on backpack: Damage minimizing humanoid fall on targeted body segment using momentum control. In *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 703–712.

[17] Javier Ruiz-del Solar, Javier Moya, and Isao Parra-Tsunekawa. Fall detection and management in biped humanoid robots. In *Proc. of 2010 IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3323–3328.

[18] Kiyoshi Fujiwara, Fumio Kanehiro, Hajime Saito, Shuuji Kajita, Kensuke Harada, and Hirohisa Hirukawa. Falling motion control of a humanoid robot trained by virtual supplementary tests. In *Proc. of 2004 IEEE ICRA*.

[19] Reimund Renner and Sven 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. IEEE, 2006.

[20] JG Karssen and Martijn Wisse. Fall detection in walking robots by multi-way principal component analysis. *Robotica*, 27(02):249–257, 2009.

[21] Oliver Höhn, J Gačnik, and Wilfried Gerth. Detection and classification of posture instabilities of bipedal robots. In *Climbing and walking robots*, pages 409–416. Springer, 2006.

[22] Kunihiro Ogata, Koji Terada, and Yasuo Kuniyoshi. Falling motion control for humanoid robots while walking. In *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pages 306–311. IEEE, 2007.

[23] Oliver Höhn and Wilfried Gerth. Probabilistic balance monitoring for bipedal robots. *The International Journal of Robotics Research*, 28(2):245–256, 2009.

[24] Shivaram Kalyanakrishnan and Ambarish Goswami. Learning to predict humanoid fall. *International Journal of Humanoid Robotics*, 8(02):245–273, 2011.

[25] Yueng Santiago Delahoz and Miguel Angel Labrador. Survey on fall detection and fall prevention using wearable and external sensors. *Sensors*, 14(10):19806–19842, 2014.

[26] Koray Ozcan, Anvith Katte Mahabalagiri, Mauricio Casares, and Senem Velipasalar. Automatic fall detection and activity classification by a wearable embedded smart camera. *IEEE journal on emerging and selected topics in circuits and systems*, 3(2):125–136, 2013.

[27] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[28] Rahim Jassemi-Zargani and Dan Necsulescu. Extended kalman filter-based sensor fusion for operational space control of a robot arm. *IEEE Transactions on Instrumentation and Measurement*, 51(6):1279–1282, 2002.

[29] M Soltani, M Bozorg, and MR Zakerzadeh. Parameter estimation of an sma actuator model using an extended kalman filter. *Mechatronics*, 50:148–159, 2018.

[30] Dennis Schüthe, Felix Wenk, and Udo Frese. Dynamics calibration and real-time state estimation of a redundant flexible joint robot based on encoders and gyroscopes. In *Informatics in Control, Automation and Robotics*, pages 385–409. Springer, 2018.

[31] Hongling Wang, Chengjin Zhang, Yong Song, and Bao Pang. Information-fusion methods based simultaneous localization and mapping for robot adapting to search and rescue postdisaster environments. *Journal of Robotics*, 2018, 2018.

[32] Fay B Horak. Postural orientation and equilibrium: what do we need to know about neural control of balance to prevent falls? *Age and ageing*, 35(suppl 2):ii7–ii11, 2006.

[33] Paul M Bays and Daniel M Wolpert. Computational principles of sensorimotor control that minimize uncertainty and variability. *The Journal of physiology*, 578(2):387–396, 2007.

[34] RJ Peterka. Sensorimotor integration in human postural control. *Journal of neurophysiology*, 88(3):1097–1118, 2002.

[35] Ronald K Pearson, Yrjö Neuvo, Jaakko Astola, and Moncef Gabbouj. Generalized hampel filters. *EURASIP Journal on Advances in Signal Processing*, 2016(1):87, 2016.

[36] T Huang, GJTGY Yang, and G Tang. A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(1):13–18, 1979.

[37] Stephen N Robinovitch, Rebecca Brumer, and Jessica Maurer. Effect of the âĂIJsquat protective responseâĂİ on impact velocity during backward falls. *Journal of biomechanics*, 37(9):1329–1337, 2004.

[38] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006.

[39] Seung-Joon Yi, Byoung-Tak Zhang, Dennis Hong, and Daniel D Lee. Online learning of low dimensional strategies for high-level push recovery in bipedal humanoid robots. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1649–1655. IEEE, 2013.

[40] K Fujiwara, F Kanehiro, S Kajita, K Kaneko, K Yokoi, and H Hirukawa. Falling motion control to minimize damage to biped humanoid robot. In *Proc. of 2002 IEEE IROS*.

[41] Kiyoshi Fujiwara, Fumio Kanehiro, Shuuji Kajita, and Hmhisa Hirukawa. Safe knee landing of a human-size humanoid robot while falling forward. In *Proc. of 2004 IEEE/RSJ IROS*.

[42] Javier Ruiz-del Solar, R Palma-Amestoy, Román Marchant, Isao Parra-Tsunekawa, and Pablo Zegers. Learning to fall: Designing low damage fall sequences for humanoid soccer robots. *Robotics and Autonomous Systems*, 57(8):796–807, 2009.

[43] Kiyoshi Fujiwara, Shuuji Kajita, Kensuke Harada, Kenji Kaneko, Mitsuharu Morisawa, Fumio Kanehiro, Shinichiro Nakaoka, and Hirohisa Hirukawa. An optimal planning of falling motions of a humanoid robot. In *Proc. of 2007 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 456–462.

[44] Qingqing Li, Xuechao Chen, Yuhang Zhou, Zhangguo Yu, Weimin Zhang, and Qiang Huang. A minimized falling damage method for humanoid robots. *International Journal of Advanced Robotic Systems*, 14(5):1729881417728016, 2017.

[45] Qingqing Li, Zhangguo Yu, Xuechao Chen, Weimin Zhang, Zhaoyang Cai, Qian Liang, Qinqin Zhou, Zelin Huang, and Qiang Huang. A falling forwards protection strategy for humanoid robots. In *ROMANSY 22–Robot Design, Dynamics and Control*, pages 314–322. Springer, 2019.

[46] Jiuguang Wang, Eric C Whitman, and Mike Stilman. Whole-body trajectory optimization for humanoid falling. In *American Control Conference (ACC), 2012*, pages 4837–4842. IEEE, 2012.

[47] Shihao Wang and Kris Hauser. Real-time stabilization of a falling humanoid robot using hand contact: An optimal control approach. In *Humanoid Robotics (Humanoids), 2017 IEEE-RAS 17th International Conference on*, pages 454–460. IEEE, 2017.

[48] Kunihiro Ogata, Koji Terada, and Yasuo Kuniyoshi. Real-time selection and generation of fall damage reduction actions for humanoid robots. In *Proc. of 2008 IEEE-RAS Humanoids*.

[49] Shuuji Kajita, Rafael Cisneros, Mehdi Benallegue, Takeshi Sakaguchi, Shin'ichiro Nakaoka, Mitsuharu Morisawa, Kenji Kaneko, and Fumio Kanehiro. Impact acceleration of falling humanoid robot with an airbag. In *Proc. of 2016 IEEE-RAS 16th Int. Conf. on Humanoid Robots*, pages 637–643.

[50] Jinoh Lee, Wooseok Choi, Dimitrios Kanoulas, Rajesh Subburaman, Darwin G Caldwell, and Nikolaos G Tsagarakis. An active compliant impact protection system for humanoids: Application to walk-man hands. In *Proc. of 2016 IEEE-RAS 16th Int. Conf. on Humanoid Robots*, pages 778–785.

[51] Sehoon Ha and C Karen Liu. Multiple contact planning for minimizing damage of humanoid falls. In *Proc. of 2015 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2761–2767.

[52] Visak CV Kumar, Sehoon Ha, and C Karen Liu. Learning a unified control policy for safe falling. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 3940–3947. IEEE, 2017.

[53] Yuhang Zhou, Xuechao Chen, Huaxin Liu, Zhangguo Yu, Weimin Zhang, and Qiang Huang. Falling protective method for humanoid robots using arm compliance to reduce damage. In *Proc. of 2016 IEEE Int. Conf. on Robotics and Biomimetics*, pages 2008–2013.

[54] Vincent Samy and Abderrahmane Kheddar. Falls control using posture reshaping and active compliance. In *Proc. of 2015 IEEE-RAS 15th Int. Conf. on Humanoid Robots*, pages 908–913.

[55] Seung-kook Yun and Anshuman Goswami. Tripod fall: Concept and experiments of a novel approach to humanoid robot fall damage reduction. In *Proc. of 2014 IEEE Int. Conf. on Robotics and Automation*, pages 2799–2805.

[56] Dingsheng Luo, Yian Deng, Xiaoqiang Han, and Xihong Wu. Biped robot falling motion control with human-inspired active compliance. In *Proc. of 2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 3860–3865.

[57] Gustavo A Cardona, Wilfrido Moreno, Alfredo Weitzenfeld, and Juan M Calderon. Reduction of impact force in falling robots using variable stiffness. In *SoutheastCon*, pages 1–6. IEEE, 2016.

[58] Seung-kook Yun, Ambarish Goswami, and Yoshiaki 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. IEEE, 2009.

[59] Umashankar Nagarajan and Ambarish Goswami. Generalized direction changing fall control of humanoid robots among multiple objects. In *Proc. of 2010 IEEE Int. Conf. on Robotics and Automation*, pages 3316–3322.

[60] Brenda E Groen, Vivian Weerdesteyn, and Jacques Duysens. Martial arts fall techniques decrease the impact forces at the hip during sideways falling. *Journal of biomechanics*, 40(2):458–462, 2007.

[61] Marianne JR Gittoes, Mark A Brewin, and David G Kerwin. Soft tissue contributions to impact forces simulated using a four-segment wobbling mass model of forefoot–heel landings. *Human movement science*, 25(6):775–787, 2006.

[62] A Ailon and R Ortega. An observer-based set-point controller for robot manipulators with flexible joints. *Systems & Control Letters*, 21(4):329–335, 1993.

[63] Sujit Nair and Naomi Ehrich Leonard. A normal form for energy shaping: application to the furuta pendulum. In *Proc. of 2002 IEEE 41st Int. Conf. on Decision and Control*, volume 1, pages 516–521.

[64] José Guadalupe Romero, Alejandro Donaire, and Romeo Ortega. Robust energy shaping control of mechanical systems. *Systems & Control Letters*, 62(9):770–780, 2013.

[65] Russ Tedrake. Underactuated robotics: Learning, planning, and control for efficient and agile machines: Course notes for mit 6.832. *Working draft edition*, pages 33–36, 2009.

[66] Kiyoshi Fujiwara, Shuuji Kajita, Kensuke Harada, Kenji Kaneko, Mitsuharu Morisawa, Fumio Kanehiro, Shinichiro Nakaoka, and Hirohisa Hirukawa. Towards an optimal falling motion for a humanoid robot. In *Proc. of 2006 6th IEEE-RAS Int. Conf. on Humanoid Robots*, pages 524–529.

[67] Nikos G Tsagarakis, Darwin G Caldwell, F Negrello, W Choi, L Baccelliere, VG Loc, J Noorden, L Muratore, A Margan, A Cardellino, et al. Walk-man: A high-performance humanoid platform for realistic environments. *Journal of Field Robotics*, 34(7):1225–1259, 2017.

[68] Rajesh Subburaman, Jinoh Lee, Darwin G Caldwell, and Nikos G Tsagarakis. Multi-sensor based fall prediction method for humanoid robots. In *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2016 IEEE International Conference on*, pages 102–108. IEEE, 2016.

[69] Rajesh Subburaman, Jinoh Lee, Darwin G Caldwell, and Nikos G Tsagarakis. Online falling-over control of humanoids exploiting energy shaping and distribution methods. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 448–454. IEEE, 2018.

[70] Robert L Jackson, Itzhak Green, and Dan B Marghitu. Predicting the coefficient of restitution of impacting elastic-perfectly plastic spheres. *Nonlinear Dynamics*, 60(3):217–229, 2010.

Glossary

List of Acronyms

CoG	center of gravity
PE	potential energy
KE	kinetic energy
3D	3-dimensional
2D	2-dimensional
EU	European Union
DoF	degrees of freedom
TIP	telescopic inverted pendulum
SPR	segmented planar robot
IMU	inertial measurement unit
FPS	foot pressure sensor
URDF	universal robot description format
ROS	Robot Operating System
RT	Real-Time
API	application programming interface
R-HAL	robot hardware abstraction layer
XDDP	cross domain datagram protocol

IPC	inter process communication
FT	force/torque
RGB-D	depth sensor
GCoM	Ground projection of the center of mass
CoM	center of mass
CoP	Center of pressure
ZMP	zero moment point
FRI	Foot rotation indicator
ZRAM	Zero rate of change of angular momentum
CP	capture point
CCP	corrected capture point
CNS	central nervous system
LiDAR	light detection and ranging
FC	foot contact
OPF	optical flow
FIV	fall indicator variables
KF	Kalman filter-based
PF	performance-based
CPF	capture point with foot contact
MDP	Markov Decision Process
CRP	critical rolling parameter
ORC	online rolling controller
EDPH	energy distribution polyhedron
EDP	energy distribution polygon
ODE	open dynamics engine
ES	energy shaping

ED	energy distribution
ES+ED	energy shaping with energy distribution
GP	Gaussian process
Rviz	ROS visualization
RTOS	Real-Time Operating System

Resume

Education

University of Genova / Istituto Italiano di Tecnologia Genova, Italy
PhD in BioEngineering and Robotics. Expected Feb 2019

Delft University of Technology Delft, Netherlands
M.Sc. in Bio Mechanical Design (BioRobotics), Average grade: 7.58 Sep. 2013-15

Government College of Engineering, Anna University Chennai, India
B.E. in Mechanical Engineering, Grade: 77% Jul. 2005 - 2009

Robert Mat. Hr. Sec. School Chennai, India
SSLC, Matric Board: 89%; PU Exam: 92% Apr 2003-05

Related Research Experience

Istituto Italiano di Tecnologia Genova, Italy
PhD Thesis: Fall Prediction and Controlled Fall of Humanoids Nov 2015 - Present

- Developed a multi-sensor based fall prediction technique to predict the fall over of humanoids reliably on different terrains.
- Proposed two novel controllers based on active energy control concepts: injection, shaping, and distribution to mitigate the damages sustained by a humanoid during its fall over.
- The proposed controllers are online applicable and adaptable to the arbitrary disturbances.

Istituto Italiano di Tecnologia Genova, Italy
M.Sc. Internship: Recovery control for Humanoids Jul-Dec 2014

- A controller for a humanoid to recover from prone to standing configuration was proposed by optimizing the contact force using inverse dynamics.
- Involves validation of optimization algorithm, design of quasi-static poses and development of a controller to transit between each pose.

Delft University of Technology Delft, Netherlands
M.Sc. Thesis: General control law for multi-limbed robots for pushing task Jan-Sep 2015

- The main scope is to analyze how humans perform the pushing task and synthesize a controller for imbibing similar techniques into multi-limbed robots.
- Given an environment and a pushing task to any multi-limbed robots, the controller will automatically enable the robot to adapt its posture and contacts for achieving the desired task with minimum effort.

Anna University Chennai, India
B.E. Project: VACROB - An Autonomous Vacuum Cleaning Robot Apr 2008-09

- Objective is to analyze and study the commercially existing IRobot's vacuum cleaning robot, ROOMBA (basic version).
- Indigenously design and develop a robot to be more efficient and cost effective than ROOMBA.

Technical Skills

Modeling Software Packages	Solid Works, Solid Edge, Catia, and AutoCAD
Mechanical Systems Analysis Software	ANSYS V13 and MSC Adams
Robot Simulation Packages	Gazebo, V-REP, and Robot Studio 5.14
Programming languages	C, C++, bash, and Java(basic)
Numerical computing environment	Matlab and python
Operating Systems	Linux, ROS, and Windows

Publications

1. Jinoh Lee and Rajesh Subburaman, *"Rolling! Energy Control Concepts for humanoid falls."* **IROS 2018 Workshop-Humanoid Robot Falling: Fall Detection, Damage Prevention, and Recovery Actions.** , Oct. 1, 2018, Madrid, Spain (invited talk)

2. Rajesh Subburaman, Dimitrios Kanoulas, Luca Muratore, Nikos G.Tsagarakis, and Jinoh Lee, "*Human Inspired Fall Prediction Method for Humanoid Robots*", **Robotics and Autonomous Systems journal.** (Submitted)

3. Rajesh Subburaman, Nikos Tsagarakis, and Jinoh Lee "*Online Rolling Motion Generation for Humanoid Falls Based on Active Energy Control Concepts*", **2018 IEEE International Conference on Humanoid Robots (Humanoids)**, November 6-9, 2018, Beijing, China.

4. Rajesh Subburaman, Jinoh Lee, Darwin G. Caldwell, Nikos Tsagarakis, "*Online Falling-Over Control of Humanoids Exploiting Energy Shaping and Distribution Methods*", **In Proc. of 2018 IEEE International Conference on Robotics and Automation (ICRA), pages 448-454**, Brisbane, Australia, May 21-25, 2018.

5. Jinoh Lee, Wooseok Choi, Dimitrios Kanoulas, Rajesh Subburaman, Darwin G. Caldwell, and Nikolaos G. Tsagarakis, "*An Active Compliant Impact Protection System for Humanoids: Application to WALK-MAN Hands*," **In Proc. of 2016 IEEE-RAS International Conference on Humanoid Robots (Humanoids 16), pages 778-785**, Cancun, Mexico, Nov. 15-17, 2016.

6. Rajesh Subburaman, Jinoh Lee, Darwin G. Caldwell, Nikos Tsagarakis,"*Multi-Sensor Based Fall Prediction Method for Humanoid Robots*", **In Proc. of 2016 IEEE/RSJ Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems (MFI), pages 102-108**, Baden-Baden, Germany, Sept. 19-21, 2016.