

Real-time Collision Detection for Position-Controlled Humanoid Robots

Alejandro Ramirez-Serrano, Shadi Moghaddasi

University of Calgary

2500 University Dr NW, Calgary, Canada

aramirez@ucalgary.ca; Shadi.Moghaddasi1@ucalgary.ca

Abstract - Humanoid robots (humanoids), at least in theory, can assist humans and work with them in cluttered and confined environments. However, further developments are needed to fully enable them to work in close proximity (even in physical interactions) with humans while not risking the safety of themselves and the objects and people around them. Current methods have not been fully successful in preparing humanoids for 100% safe physical Human Robot Interaction (HRI) due partially to the unresolved challenges of detecting the characteristics of the surrounding environment. Furthermore, current humanoids employ expensive and fragile equipment making them costly, thus limiting humanity of using them. This paper presents a novel real-time and hardware inexpensive collision detection methodology that employs signals from the robots' motor joints and data processing capabilities from the computers running the robot. The approach enables the safe close-proximity HRI for position-controlled humanoids that minimizes any negative effects caused by the detected collision. Using the proposed algorithm, humanoids can speedily identify the joint(s) responsible for the collision and the affected joints from which effective path planning movement control can be determined. Experimental results using a life-size humanoid robot having 29 degrees of freedom are presented that demonstrates the applicability of the proposed approach.

Keywords: Humanoids, Safe Human Robot Interaction (HRI), Collision Detection, Signal Analysis, Path Planning

1. Introduction

Human beings have been envisioning having strong, agile, and trustworthy human-alike companionable machines for more than a century. The expectations of fully realizing such artifacts was initially depicted in the form of cartoon characters such as “*Rosie the Robot Maid*” during the 1960's TV show “*The Jetsons*”. Today, depending on the application, humanoids come in different size, dexterity, and Human-Robot Interaction (HRI) capabilities. Due to the inability of current humanoids to deal with our chaotic highly dynamic and dense industrial (e.g., manufacturing floors), recreational (e.g., sporting events), and other (e.g., farmer markets) environments these robots have been mainly deployed in engineered static environments. In the medical industry for example, where safety is paramount, humanoids require the presence of a human operator for the safe execution of diverse maneuvers. In industry, relatively large robots are kept within safe distances from humans to prevent accidents and increase safety [1]. However, if humanoids are to be of benefit to society and meet the expectations of having trustworthy companions/servants, effective methodologies are needed to enable autonomous systems to deal with the high dynamics and frequent changes of our world (e.g., obstacle avoidance of high-speed entities) or develop methodologies to minimize the negative effects of unavoidable (accidental) collisions. The paper focus on increasing safety during the deployment and operation of humanoids by minimizing the effects of collisions on the robot, the user(s) and the environment as it is not a question if collisions will occur but rather it is a fact that collisions will occur. In the context of this paper the following three aspects are considered when discussing the safe operation of humanoids within our human-centric world:

- The robot's ability to operate amongst and within close proximity of humans and objects without *unplanned collisions*.
- Robots capable of safely maneuvering in cluttered and dynamic confined spaces typically designed for humans.
- Robots having the mechanisms to avoid causing damages to itself, humans, and the objects around it.

Considering the above-mentioned aspects, we argue that the interacting mechanisms among such qualities increases safe HRI. For this we focus on *unplanned* or *undesired collisions* as well as *planned collisions* which are defined as:

Planned collision is defined as an expected force impact from the robot's body to the environment/objects. In this paper it is desired to detect planned collisions as by properly controlling and being aware of such impacts enhances safer HRI.

Unplanned collision, a.k.a. accidents, are when two or more bodies exert a force, $F \geq x$, on each other that causes an

undesirable change in velocity/position on at least one entity (with or without damage).

Based on this, the following problem statement is formulated:

Develop a fast (instantaneous) planned & unplanned collision-detection algorithm to enable humanoids to safely maneuver in arbitrary cluttered dynamic environments and operate safely in any HRI activity.

2. Background

Some of the important aspects of a humanoid's safe operation, its ability to sense, balance, and react to errors rely on its ability to move within the environment. Pedal walking has showed to be more agreeable with the terrains that humans generally walk on [2-4]. Moreover, with the employment of legged over wheeled locomotion (using less than 3 wheels), a lower chance a humanoid has of falling or colliding on the surrounding objects [5]. Therefore, pedal gaits contribute to a safer robotic manoeuvre amongst humans. Legged locomotion on humanoid robots like ASIMO and BIPER having superior means of walking and maintaining a balanced posture when combined with suitable walking pattern generation has shown to be suitable for preventing collisions [6]. As the aspect of collision avoidance based on the perceived information about the surrounding environment highly allows a safer walking, it's beneficial to target collision avoidance and step generating in cluttered environments [7-8]. To do so most humanoids employ vision system to examine the environment and prevent collisions. Although suitable for static spaces, due to its high computer demand, computer vision is limiting when used in high-speed changing dynamic cluttered spaces where multiple objects move in somewhat chaotic fashion (like seen in a busy downtown street).

It is a well-known fact that human beings can rely on their hands to protect themselves in case of falling and have a more balanced posture when compared to humanoids. Due to humanoids' close physical characteristics to humans they can also benefit from using their hands and other body parts besides their feet for balance enhancement [7-11]. With improved and comprehensive methodologies to maintain balanced, humanoids will reach safer close-proximity interactions with humans. However, to fulfil this vision humanoids need to engage complementary technologies and methods to their current sensor and motion systems to assure effective handling of systematic errors/disturbances [7].

It is known that humans use their sense of touch to compensate for errors associated with their visual and hearing systems. Humanoids can also benefit from the contribution of their tactile sensing for higher accuracy during the overall balancing and interaction operations [11-13]. While enhancing humanoids balance and interaction mechanisms increases the possibility of close-proximity human-robot interaction, it also contributes to the aspect of physical HRI's safety. It is undeniable that a more balanced robot having numerous complementary methods to compensate for possible unforeseen errors and dynamics can increase safety. Not only robots will be able to guarantee its surrounding objects' and humans' safety but also they will prevent damaging their own self, hence not endangering its own safety [14]. Overall the employment of tactile sensing in a broad sense seems to enhance safe humanoid robot interaction [15]. Currently humanoid robots use tactile, and force sensors which among other things are used for effective pedal balancing purposes [11-13]. There are some humanoid robots that are equipped with tactile sensors in their hands and other body parts [13]. The utilization of such sensors all over the humanoid's body is thought to enable the robot to have tactile sensing comparable to a human and thus, a higher possibility of performing safe manoeuvres. However, it negatively affects the robot's computation and reaction time while increasing its costs. In addition, these sensors require high maintenance due to their complexity and fragility [16]. As a result, the employment of complementary techniques that could provide similar information to the robot without the associated complexities can be highly beneficial. Surely such robots would be able to use their tactile sensation to contribute to their balance and interaction system while not causing high expenses or a burden on the computational time for fast, safe, real-life interactions.

3. Proposed Solution: Collision Detection Algorithm

As mentioned in the Section 2, humanoid robots are, at least in theory, capable of safe operation through arbitrary cluttered environments, navigate on rough dynamic terrains, and perform effective work among and cooperating with humans. Many of the different techniques used to enable HRI, and safe humanoid robots' operation, either use visual

perception or benefit from other sophisticated sensing equipment such as tactile sensors to feedback information to the robot so that it can effectively maneuver and prevent dangerous collisions. Even though the techniques associated with visual perception and the associated sensors have been used in the field of robotics longer than tactile sensors, they are yet to be completely error-free and reliable. However, the sensors which inform the robot about its physical connection with its surrounding (e.g., tactile sensors) also help to compensate for visual perception errors. However, the family of tactile sensors (tactile, touch, and force sensors) also have shortcomings. Thus, in order to fully enable safe, reliable, dependable, friendly, and provide effective inexpensive robots, it is favorable to minimize the addition of sensing and hardware equipment and use the robot’s available hardware with advanced control, data analysis, and motion planning techniques. This will maximize the potential use of the available systems and components. Although many and diverse of the current robot characteristics can be thought to be required to enable 100% error-free and safe operation (e.g. perception, motion control, and path planning), one of the goals is to prevent collisions and minimize any negative effects that any unplanned collision could cause. However, for this to be possible in real world activities (in part because the world is highly dynamic) robots must be able to adapt.

Based on this and within the context of this work, this paper presents a fast (instantaneous) collision-detection algorithm applicable to position-controlled humanoid robots which enables them to safely maneuver in arbitrary cluttered environments and operate safely in any HRI activity. The goal of the proposed methodology is to enable humanoids to detect planned and unplanned collisions within a given range of applied force and determine (as precise as possible) the location on where the collision occurred so that effective and safer robot activities can be constructed. This is to be achieved without adding sensors or other devices to the basic available infrastructure (e.g., motors, IMU, F/T sensors, cameras) present in typical humanoids.

Figure 1 illustrates an overall flowchart of the proposed methodology for detecting collisions on the humanoid’s body. The approach consists of four layers (Fig. 1): *Perception*, *Evaluation*, *Identification*, and *Execution* Layers. The solution uniquely combines prior published work in perception and develops novel Evaluation, and Identification layers. These two layers process the received signals from the robot’s joint motors and keeps track of the controlled computer commands used during the planned maneuvers. The output is the recognized collisions (if any occurred) and their locations on the humanoids’ body. As can be observed from Figure 1 the methodology is performed in a sequential manner. A process that is repeated at every control cycle δt_m (the period that the robot’s controlling mechanism takes to update the robot’s (current) state).

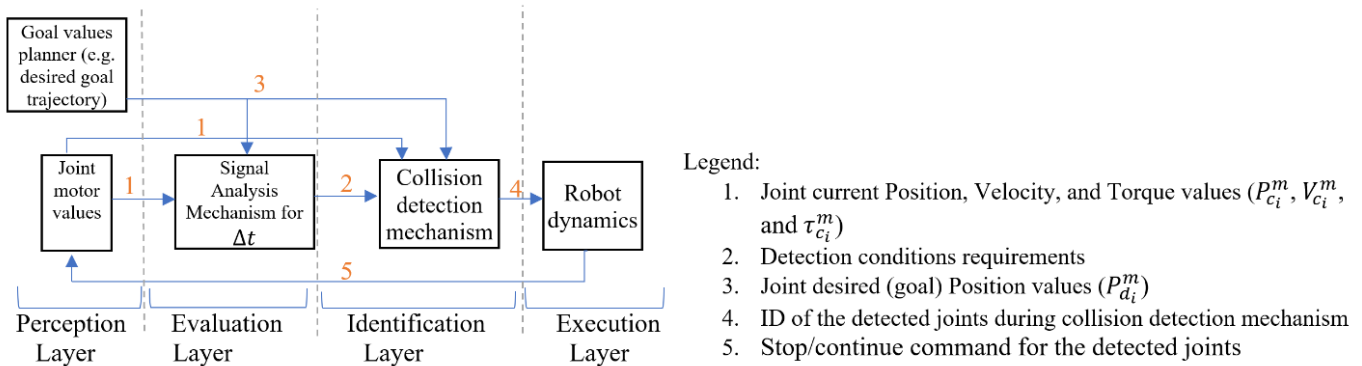


Figure 1: Flowchart of the Collision Detection Algorithm.

In brief, the proposed algorithm works as follow: The current and desired state of every joint comprising the humanoid are gathered by the Perception layer. The joint state is herein defined by the following three parameters: *a*) position, *b*) velocity, and *c*) torque. The current state of all joints is then passed to the Evaluation layer which records a sequence of joint states over a pre-defined time window. This is performed for every joint either if the joint is moving or static.

Since the sensed joint state values might have errors the goal of the Evaluation layer is to remove the erroneous values denoted herein as outliers. For this, the Evaluation layer records a set of incoming data repetitive over a short time window

$\Delta t = \sum_{m=1}^l \delta t_m$ (m refers to the number of storing iterations and Δt is the storing time window). Once the outliers (errors) are removed from the joint states data received a good set of data set is generated from where the max and min values for each parameter of interest is obtained (i.e., *max* & *min* position/velocity/torque for joint “ i ”). Herein the corresponding difference between the max & min is referred to as the parameter range for joint “ i ”. Such range is then passed to the Identification layer for processing (Section 3.2.1). Subsequently, the so-called safe range, the robot’s kinematics, and the desired goal values for each joint are used to compute if several conditions are met. The developed conditions are used to identify collisions as well as the affected robotic joints. This is a task performed by the Identification Layer which executes its task every δt_m (Section 3.1). Consequently, the Identification layer detects collisions in accordance to the results from the Evaluation layer and the desired and current joint values from the Perception layer. The Evaluation layer then helps to decide on whether to continue a specific robot maneuver (depending if a planned or unplanned collision was detected). The result of such a decision is then carried by the Execution layer, which sends the result of the decision back to the corresponding robotic joint(s) to stop or continue the on-going maneuver. If the overall algorithm does not detect a collision, the above cycle continues simultaneously as the robot performs any given maneuver. In the following sub-sections, each of the proposed layers is described in detail followed by illustrative results in Section 5.

3.1. Perception Layer

The perception layer is developed under the assumption of using position-controlled humanoid robots which operate by computing and propagating the desired future joint position values. Furthermore, since the purpose of the proposed algorithm is mimic the information provided by tactile/skin sensors, the motion information perceived from the robot’s joint motors and the values computed by the motion mechanism regarding the future (planned) states are considered as the main element (sensing information) of the Perception layer. The motor information used in the proposed approach includes the position, velocity, and internal applied torque (or force) of all the robot’s joints (Figure 1). The current and future joint states data is always utilized in the execution of the proposed collision detection algorithm. The joint values are updated every control cycle δt_m , where δt_m is the time that one cycle of the control algorithm takes to compute its future state. The δt_m is also used to represent the m^{th} iteration of the robot’s control cycle. The current state of the joints and the goal (calculated) joint values are then independently fed to the Evaluation layer for further evaluation and interpretation.

3.2. Evaluation Layer

Throughout this work, the desired (computed) position of a joint i , based on following a given trajectory at iteration (control) cycle m is denoted as $P_{d_i}^m$. Likewise, the current motor state values directly received from the motors (position, velocity and torque) for each joint i are denoted as $P_{c_i}^m$, $V_{c_i}^m$, and $\tau_{c_i}^m$, respectively. The full set of motor state values is then defined as: $S = \{P_{c_i}^m, V_{c_i}^m, \tau_{c_i}^m, P_{d_i}^m\}$ where P_{c_i} , V_{c_i} , τ_{c_i} , P_{d_i} (current values for position, velocity, torque, and desired position) are arrays of size based on the number of motors comprising the robot. As illustrated in Figure 1, the subset of S of the desired and current position, velocity, and torque values for the robotic joints is stored for each time window $\Delta t = \sum_{m=1}^l \delta t_m$.

3.2.1. Moving Filtering Mechanism

Although there are numerous available methods for interpreting erroneous data with considerable amounts of false irregularities, developing methods which result in higher accuracies in terms of detecting and replacing outliers without noticeable data loss in a time efficient manner is needed. With the purpose of reducing time and increasing computation efficiency, a dynamic filter for the current values of torque τ_{c_i} is proposed in combination with the position and velocity data to detect collisions. The proposed filter is a moving (dynamic) filtering method used to detect local outliers comprising the sensed torque values according to a window size W_s . The method identifies outliers as the data elements which have a value greater than three local scaled Median Absolute Deviation (MAD) from the local median over the sliding window having size W_s . For joint i during Δt the scaled MAD is defined as:

$$\text{Scaled MAD} = C \times \text{median} (|\tau_{c_i}^m - \text{median}(\tau_{c_i})|) \quad (1)$$

where

$$C = \frac{-1}{\sqrt{2} \times \text{erfc}^{-1}(\frac{3}{2})} \quad (2)$$

In Eqn. 2, erfc^{-1} stands for the Inverse complementary error function which provides the following value: $C=1.4826$. Once all outlier data points comprising each array of τ_{c_i} are identified, each outlier is replaced with a modified value. The outliers are replaced with the computed threshold values using the “linear interpolation neighboring” method. The resulting filtered array (no outliers present) for the torque is then denoted as $\hat{\tau}_{c_i}$ (filtered current torque values).

3.2.2. Generation of Collision Detection Mechanism Requirements

Simultaneously with the filtering mechanism, the *max* and *min* values for P_{c_i} and its differential value are computed and stored in a matrix, $MAM_{P_{c_i}}$. Subsequently, for each pair of positions ($P_{c_i}^m, P_{d_i}^m$) the difference between the current, $P_{c_i}^m$, and desired, $P_{d_i}^m$, values for time period Δt are computed. The differentiated values of desired and current position are computed per Eqn. 3 and stored into new array, dP_i .

$$dP_i = |P_{c_i} - P_{d_i}| \quad (3)$$

Thereafter, the max values for each array of dP_i is calculated and stored as dP_i^{Max} . The values of dP_i^{Max} are updated during each Δt and used during the collision detection process (Eqns. 6 and 7).

The third and last condition used to identify collisions uses the slopes between the $\hat{\tau}_{c_i}^m$ variables. During diverse tests it was observed that even though the torque values might follow a constantly rising or falling pattern between each $\hat{\tau}_{c_i}^m$, the absolute values of the slopes between each $\hat{\tau}_{c_i}^m$ for each period δt_m can be effectively utilized for detecting collisions. For this purpose, Eqn. 4, is used to find the $m - 1$ absolute slope values between $\hat{\tau}_{c_i}^m$ variables during the control cycle δt_m for each joint i :

$$S_{\hat{\tau}_{c_i}^{m-1 \times 1}} = \left\| \left[\begin{array}{ccc} \frac{\hat{\tau}_{c_i}^1 - \hat{\tau}_{c_i}^2}{\delta t_m} & \frac{\hat{\tau}_{c_i}^2 - \hat{\tau}_{c_i}^3}{\delta t_m} & \dots & \frac{\hat{\tau}_{c_i}^{m-1} - \hat{\tau}_{c_i}^m}{\delta t_m} \end{array} \right] \right\| \quad (4)$$

From Eqn. 4 it is observed that $S_{\hat{\tau}_{c_i}}$ is an $m - 1 \times 1$ matrix calculated for the time period Δt (i.e., $S_{\hat{\tau}_{c_i}}$ includes the slopes between torque values during the time Δt). Subsequently, the maximum value within $S_{\hat{\tau}_{c_i}}$ is identified as $S_{\hat{\tau}_{c_i}}^{Max}$. The values of $S_{\hat{\tau}_{c_i}}^{Max}$, updated for each Δt , are used for the third and last condition (Eqn. 7) for detecting collisions.

3.3. Identification Layer

In this sub-section, the previously explained requirements are utilized to generate a time and computational efficient collision detection for humanoid robots. At the start of the process the proposed algorithm undergoes a startup (initialization) time of Δt . This is due to the minimum time the Evaluation Layer requires to store, filter, and process its input data. Once the algorithm has finished its startup time, for each δt_m , defined as the time between each array of received data, the received values for the present time m and the previous time iteration $m - 1$ are compared with the previously computed (during the preceding Δt) $MAM_{P_{c_i}}$. This is performed for each joint i using Eqn. 5 where $P_{c_i}^{m-1}$ stands for the preceding time iteration $m - 1$ position values.

$$|P_{c_i}^{m-1} - P_{c_i}^m| \leq MAM_{P_{c_i}} \quad (5)$$

The values of dP_i^{Max} are used for developing the second condition for detecting collisions. According to Section 4.2.2, the absolute differential value for the current time (iteration m) received and desired position ($P_{c_i}^m$ & $P_{d_i}^m$) are compared with dP_i^{Max} . Such comparisons are computed via Eqn. 6 during Δt for each joint. If the absolute between the current (iteration m) and the desired values is less or equal than the maximum of the absolute differential value computed between the received and desired values in the preceding Δt period, then it is identified that a collision not been detected. Meeting the 2nd condition alongside the 1st and 3rd conditions provides supporting information for accurately detecting collisions.

$$|P_{c_i}^m - P_{d_i}^m| \leq dP_i^{max} \quad (6)$$

In Section 3.2.2 the requirements for developing the last collision detecting condition were described. According to Equation 3.4, at each control cycle δt_m , the absolute slope value between the current time m and the previous cycle (time $m - 1$) for each joint i is calculated and compared with the value obtained from Eqn. 4 ($S_{\hat{\tau}_{c_i m-1 \times 1}}$) where $\tau_{c_i}^{m-1}$ represents the torque values for each joint i for the preceding time iteration $m - 1$.

$$\left| \tau_{c_i}^{m-1} - \tau_{c_i}^m / \delta t_m \right| \leq S_{\hat{\tau}_{c_i}}^{Max} \quad (7)$$

If the above three conditions are met (i.e., Equations 5, 6 and 7) is said that a collision has not taken place. The results for the 3 conditional statements are published for all joints i as binary values 0 (no collision has been detected) and 1 (collision detected) in addition to the current readings from the robotic joints $P_{c_i}^m$, $V_{c_i}^m$, and $\tau_{c_i}^m$, and the joints' ID number from which the joints where a collision has been detected can be rapidly identified. Hence, for a humanoid robot with $i = 1, 2, \dots, n$ joints the following three results (Eqn. 8) are obtained where DMC_j , $DJID_j$, and DJ_{C_j} represent the “Detection Matrix”, “Detected Joints' ID”, and “Detected Joints' current readings” for the j th condition ($j = 1, 2, 3$), respectively.

$$DMC_j = [0|1 \quad 0|1 \quad \dots \quad 0|1]_{i \times 1}, \quad (8a)$$

$$DJID_j = [0|1 \quad 0|2 \quad \dots \quad 0|i]_{i \times 1}, \text{ and} \quad (8b)$$

$$DJ_{C_j} = \begin{bmatrix} \tau_{c_1}^m | 0 & P_{c_1}^m | 0 & V_{c_1}^m | 0 \\ \vdots & \vdots & \vdots \\ \tau_{c_i}^m | 0 & P_{c_i}^m | 0 & V_{c_i}^m | 0 \end{bmatrix}_{i \times 3} \quad (8c)$$

The results obtained from Eqn. 8 are then combined using the logical operator “AND” resulting is a set of three matrices generated for a given humanoid robot having n joints: $i = 1, 2, \dots, n$ where the terms $FDMC$, $FDJID$, and FDJ_C represent the “Final Detection Matrix”, “Final Detected Joints' ID”, and “Final Detected Joints' readings”, respectively.

$$FDMC = (DMC_1 \wedge DMC_2 \wedge DMC_3), \quad (9a)$$

$$FDJID = (DJID_1 \cap DJID_2 \cap DJID_3), \text{ and} \quad (9b)$$

$$FDJ_C = (DJ_{C_1} \cap DJ_{C_2} \cap DJ_{C_3}) \quad (9c)$$

Since the values for the exceeding torque, position and velocity values are identified in Eqn. 9, the proposed methodology finds the approximated value for the applied torque on the robotic joints as well as the amount of displacement and velocity change. The results of the Identification layer are passed to the Execution layer which

completes the operation architecture. At this stage, based on the results of the collision detection process the robot can stop or continue a given maneuver.

4. Experimental Results

To test and analyse the proposed collisions detection algorithm the process was implemented in a life-size (5.5 feet tall) experimental humanoid robot having 29 degrees of freedom (servomotor joints). A set of representative results is provided in Figures 2 & 3. The robot was subjected to a series of collisions executed with a Stanley 51-104 rubber mallet at various locations within the robot's body. To not damage the robot or its joints the applied collision forces used were in line with the targeted joints. That is, the applied force of each hit was below the torque threshold of each joint. The control cycle δt_m was set to 0.008 seconds and the values for S were stored for twenty data captured cycles, $m = 20$. Thus, the algorithm captured motor sensor data for a timer period of $\Delta t = 0.16$ [s]. Therefore, the collision detection mechanism started to detect collisions 0.16 [s] after the initialization of the robot's desired and current joints data publishing. For the used storing data number of 20, the Window Size $W_s=9$ was used.

Figures 2 illustrates the results of a light intensity collision on the more sensitive group of 20 [Watt] servomotors. The tests were conducted on two cases: i) robot's joints idle, and ii) moving. The right 2 columns of the figure show the recorded time when a collision was detected (after the collision took place) and the corresponding location on the body. The proposed approach was able to detect the collisions on the targeted area as well as on the joints proximal to the place where the collision took place. Collision on some of the joints were detected as a result of torque propagating through the robot's body caused by the collision (neck and left hand joints for the idle experiment and right hand and torso for the head moving experiment).

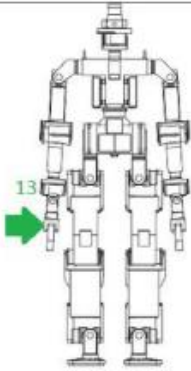
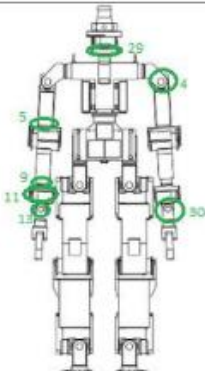
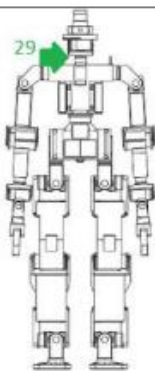
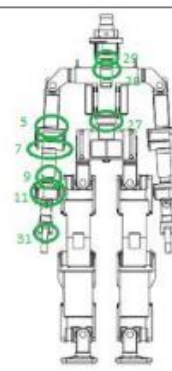
Points of Collision	Test condition	Joints detecting the collision	Affected joints due to the collision (ID)	Detection time after the collision (Seconds)
	Robot state: Idle Impacted joint ID: 13 Servomotor power: 20 W Closest joints to the collision: 11 and 9 Collision force: Light intensity		29	0.188, 0.22
			4	0.236
			5	0.34
			13	0.356
			9	0.424, 0.472, 0.52
			11	0.432, 0.464, 1.152
			30	0.916
	Robot state: Moving (moving head side to side) Joints involved in the maneuver: neck Impacted joint ID: 29 Servomotor power: 20 W Closest joints to the collision: 29 and 28 Collision force: Light		28	0.242, 0.258, 0.274, 0.29, 0.322
			29	0.222
			7	0.35
			5	0.374
			31	0.374
			9	0.398
			11	0.406, 0.422
			27	0.238, 0.254

Figure 2: Real-time collision detection of the humanoid’s 20 W servomotor joints.

During a series of more complex experiments, various groups of servomotors were hit using various intensity collisions. The applied collisions shown in Figure 3 were performed one at a time in a relatively fast succession. For the results shown in Figure 3 three collisions were applied (right wrist, left hip, and right ankle). The proposed method was successful in detecting the areas of the collision as well as the torque propagation resulted from the collisions. Some of the more sensitive servomotor joints like joint ID 29 (neck) erroneously detected the collisions which can be explained similarly to the results in Figure 2.

Points of Collision	Test condition	Joints detecting the collision	Affected joints due to the collision (ID)	Detection time after the collision (Seconds)
	Robot state: Idle		13	0.02
	Impacted joints ID: 13, 16 and 25		5	0.028
	Servomotor power: 20, 100 and 200 W		29	0.108, 0.124
	Closest joints to the 1 st collision: 13		14	0.148, 0.164, 0.18, 0.196, 0.26
	Closest joints to the 2 nd collision: 18		12	0.164, 0.18
	Closest joints to the 3 rd collision: 23		18	0.196
	Collision force: Light, Medium, and Hard (in order)		16	0.204
			13	0.26
			10	0.284
			11	0.276, 0.308
	21	0.324, 0.548		
	23	0.332, 0.348		
	25	0.372		
	18	0.42		

Figure 3: Real-time collision detection of the humanoid’s servomotor joints in idle state.

6. Conclusions

From the numerous experimental results, it is concluded that the proposed collision detection algorithm is able to successfully detect collision on the humanoid’s body as fast as 0.02 seconds (Figure 3) from the time the corresponding collision took place. On average (based on 60 diverse tests), the proposed algorithm takes 0.16 seconds to analyse the received signals from the robotic joints and can identify which joints where impacted as fast as 0.008 seconds. Furthermore, the location of the collision in addition to the other impacted areas could be found very quickly. Using the proposed approach, humanoid robots can detect and locate collisions on their body which should enable them to work near humans and even engage in physical HRI activities in a safe manner. This should be even possible despite the possibility of failures in the robot’s visual perception and movement algorithms. Thus, the proposed approach should be able to provide a tool to minimize the associated costs and disadvantages of current expensive and delicate tactile sensing sensors and their mechanisms. Despite the developments presented, there are numerous improvements that can be made to the proposed algorithm including increasing the accuracy of the collision detections and developing algorithms to identify the difference between a collision and the effects of a collision propagating through the robot’s body.

References

- [1] A. M. Zanchettin, N. M. Ceriani, S. Member, P. Rocco, H. Ding, and B. Matthias, “Safety in Human-Robot Collaborative Manufacturing Environments : Metrics and Control,” vol. 13, no. 2, pp. 882–893, 2016.
- [2] A. Mathematics and C. Engineering, “Human body motion controll for humanoids and bipedal robots,” vol. 119, no. 15, pp. 705–713, 2018.
- [3] L. Li, S. Jiang, F. Dai, and X. Gao, “Dynamic model and balance control of two-wheeled robot with non-holonomic

- constraints,” *Proc. World Congr. Intell. Control Autom.*, vol. 2015-March, no. March, pp. 503–508, 2015.
- [4] R. Niiyama and Y. Kuniyoshi, “A Pneumatic Biped with an Artificial Musculoskeletal System,” *4th Int. Symp. Adapt. Motion Anim. Mach.*, pp. 80–81, 1998.
- [5] Qiang Huang, S. Kajita, N. Koyachi, K. Yokoi, H. Arai, K. Komoriya, and K. Tanie, “A high stability, smooth walking pattern for a biped robot,” *IEEE Int. Conf. Robot. Autom.*, vol. 1, no. May, pp. 65–71, 1999.
- [6] H. Miura and I. Shimoyama, “Dynamic Walk of a Biped,” *Int. J. Robot Research*, vol. 3, no. 2, pp. 60–74, 1984.
- [7] A. C. Hildebrandt, R. Wittmann, F. Sygulla, D. Wahrmann, D. Rixen, and T. Buschmann, “Versatile and robust bipedal walking in unknown environments: real-time collision avoidance and disturbance rejection,” *Auton. Robots*, no. 0123456789, 2019.
- [8] K. Sabe, M. Fukuchi, J.-S. Gutmann, T. Ohashi, K. Kawamoto, and T. Yoshigahara, “Obstacle avoidance and path planning for humanoid robots using stereo vision,” vol.1, 2004, pp. 592-597.
- [9] P. Michel, J. Chestnutt, S. Kagami, K. Nishiwaki, J. Kuffner, and T. Kanade, “GPU-accelerated real-time 3D tracking for humanoid locomotion and stair climbing,” *IEEE Int. Conf. Intell. Robot. Syst.*, 2007, pp. 463–469.
- [10] Y. Goan, K. Yokoi, Neo Ee Sian, and K. Tanie, “Feasibility of humanoid robots stepping over obstacles,” vol. 22, no. 5, 2005, pp. 130–135.
- [11] F. Abi-Farraj, B. Henze, C. Ott, P. R. Giordano, and M. A. Roa, “Torque-Based Balancing for a Humanoid Robot Performing High-Force Interaction Tasks,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, 2019, pp. 2023–2030.
- [12] V. Duchaine, N. Lauzier, M. Baril, M. Lacasse, and C. Gosselin, “A Flexible Robot Skin for Safe Physical Human Robot Interaction,” 2009, pp. 3676–3681.
- [13] E. Dean-leon, J. R. Guadarrama-olvera, F. Bergner, and G. Cheng, “Whole-Body Active Compliance Control for Humanoid Robots with Robot Skin ICS-HI,” 2019, pp. 5404–5410.
- [14] M.W. Strohmayer, “The DLR Artificial Skin Step I: Uniting Sensitivity & Collision Tolerance,” 2013, pp. 1012–1018
- [15] V. J. Lumelsky, V. J. Lumelsky, S. Member, and E. Cheung, “Whole-Sensitive Robot Arm Manipulators Real-Time Collision Avoidance in Teleoperated Whole-Sensitive Robot Arm Manipulators,” 1993, vol. 23, no. 1.
- [16] A. Yamaguchi and C. G. Atkeson, “Recent progress in tactile sensing and sensors for robotic manipulation : can we turn tactile sensing into vision ? ABSTRACT,” 2019, vol. 1864.