PROPOSAL:
LEARNING DYNAMIC MOTOR SKILLS FOR VIRTUAL AND REAL HUMANOIDS

A Thesis Proposal
Presented to
The Academic Faculty

by

Sehoon Ha

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Computer Science

Georgia Institute of Technology
April 2014

PROPOSAL:
LEARNING DYNAMIC MOTOR SKILLS FOR VIRTUAL AND REAL HUMANOIDS

Approved by:

Professor C. Karen Liu, Advisor
School of Computer Science
Georgia Institute of Technology

Professor Greg Turk
School of Computer Science
Georgia Institute of Technology

Professor Jarek Rossignac
School of Computer Science
Georgia Institute of Technology

Professor Jun Ueda
School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: 1 July 2010

SUMMARY

As the technology of computer animation and robotics advances, controlling highly dynamic motions has been a great milestone for both virtual and real humanoid characters. However, developing controllers for dynamic motor skills is still a challenging problem, which usually requires substantial amount of design effort and optimization time due to its complexity. In this proposal, we introduce a set of techniques for designing controllers for various dynamic motor skills, such as jumping, rolling, vaulting, and landing, for virtual characters and real robots.

First, we introduce new algorithms to generate falling and landing motions, which are essential motor skills to ensure the safety of humans and robots. We propose an online algorithm to control falling and landing motions of virtual characters from a wide range of heights and initial speeds, which can potentailly cause huge damages. With the suggested algorithm, we demonstrate that simple momentum planning with the proxy model of inertia and feedback-based rules can generate the natural landing motions without large stresses on the joints. Inspired by the falling control of a virtual character, we develop an optimization algorithm for multi-contact falling motions of a real robot for minimizing the damage at the impact. Unlike the existing techniques that usually consider the desired contacts as invariant features, we search over a sequence of contacts to find the best falling motion that can handle a wider range of situations.

Second, we propose a human-in-the-loop (HITL) system to develop dynamic controllers under the guidance of a human coach. In this system, the user can provide a sequence of high-level instructions to iteratively train dynamic controllers of characters as if coaching a human trainee. To facilitate the mapping between high-level

instructions and control variables, we introduce a new representation of dynamic controllers, the “motor tree”, which also enables flexible re-assembly and efficient re-optimization by preserving the invariant features of motor skills. Further, the optimization process is accelerated by utilizing the failed previous trials. By considering noise and uncerntainty from real hardwares, we also train a real robot to perform dynamic motor skills such as rolling using our proposed system.

By incorporating the proposed techniques, we can produce highly-dynamic motions of virtual characters and real robots.

CHAPTER I

INTRODUCTION

Performing highly dynamic motions with agility and grace has been one of the greatest challenges in sports, computer animation, and robotics. A wide variety of athletics, such as acrobatics or free running, demonstrate the efficient and artistic movements that involve abrupt changes of momentum and contacts. Furthermore, these motor skills are transferred to virtual characters in animations and games to express the intention of designers and react to user interactions. Robotics, another application of dynamic controllers, also started to tackle the agile movements and demonstrated running, jumping, and landing motions with real hardwares. Despite the recent progress, learning dynamic motor skills still remains a very difficult problem because it needs to execute the task with great agility, ensure safety, and demonstrate self-expression.

In fact, developing dynamic controllers for virtual characters and real robots can be considered related problems, which can benefit each other. Since the control problems in two domains have shared properties, such as non-linearity, high-dimension, and discontinuity, an algorithm developed in one domain can be transferred to the other domain. However, control of real robots is more constrained due to the sensor uncertainty and hardware limitations, which usually require more robustness than control of virtual characters. Therefore, developing an algorithm in virtual environment to prove its full capability and transferring it to real hardwares would be a promising research direction, which is adopted in this proposal.

In this proposal, we introduce a set of techniques to expedite the learning process of dynamic controllers for various dynamic motor skills. The rest of the sections are organized as follows:

- **Optimization of Falling and Landing motions** In this proposal, we tackle the problem of controlling safe falling and landing motion for virtual characters and robots, which is a fundamental motor skill because highly dynamic motions involve the abrupt changes of contacts and can cause huge damages on the body parts. While absorbing the shock at the impact, a successful landing controller also should be able to maintain readiness for the next action by managing the momentum properly. For the virtual character, we introduce a fast and robust optimization algorithm for controlling falling and landing motions of virtual characters from a wide rage of heights and initial speeds. while reducing joint stress. Further, we develop a safe falling algorithm for a robot by planning a sequence of multiple contacts, which endures larger external perturbatations comparing to the existing methods.

- **Human-guided iterative training of dynamic motions** Also, we introduce an iterative training system for dynamic motor skills inspired by human coaching techniques, which uses human-in-the-loop (HITL) optimization for interactive training. In this system, the user only needs to provide a primitive initial controller and high-level, human-readable instructions as if coaching a human trainee. The virtual character interprets the provided instructions, accumulate the knowledge from the human coach, and iteratively improves its motor skills by optimizing control parameters. We propose a new hierarchical representation of controllers, the "motor tree" for interpreting the instructions and accumulating the knowledge on motions. In addition, we develop a new sampling-based optimization method, Covariance Matrix Adaptation with Classification (CMA-C), which efficiently solves the constrained problem by estimating the infeasible region from bad samples. The system is further extended by considering the hardware limitations and uncertainties to support training dynamic motions for robots. With the proposed system, we demonstrate the design process of comlex dynamic controllers including jumps, vaults, rolls, and flips.

CHAPTER II

OPTIMIZATION OF FALLING AND LANDING MOTIONS

This section describes algorithms for generating natural and safe falling and landing motions of virtual and real humanoids. In the prior project, we developed an online algorithm for simulated characters to generate natural falling and landing motions from different heights and initial conditions, while absorbing impact. In addition, we investigate a scenario of safe falling strategy for robots to protect themselves from large external perturbations by executing breakfall techniques.

2.1 Prior Work: Falling and Landing Motion Control for Character Animation

Figure 1: A simulated character lands on the roof of a car, leaps forward, dive-rolls on the sidewalk, and gets back on its feet, all in one continuous motion.

In our prior work [14], we introduce a new method to generate agile and natural human landing motions in real-time via physical simulation without using any mocap or pre-scripted sequences. We develop a general controller that allows the character to fall from a wide range of heights and initial speeds, continuously roll on the ground, and get back on its feet, without inducing large stress on joints at any moment (Figure 1). The character's motion is generated through a forward simulator and a control algorithm that consists of an airborne phase and a landing phase. During the airborne phase, the character optimizes its moment of inertia to meet the ideal relation between the landing velocity and the angle of attack, under the laws of conservation of momentum. The landing phase can be divided into three stages: impact, rolling, and getting-up. To reduce joint stress at landing, the character leverages contact forces to control linear momentum and angular momentum, resulting in a rolling motion which distributes impact over multiple body parts. We demonstrate that our control algorithm can be applied to a variety of initial conditions with different falling heights, orientations, and linear and angular velocities. Simulated results show that our algorithm can effectively create realistic action sequences comparable to real world footage of experienced freerunners.

2.2 Multi-contacts Falling Motion Control for a Humanoid Robot

2.2.1 Problem Description

In this section, we propose to develop a safe falling controller for humanoid robots, which ensures the safety of the robots from the large external perturbations. Our approach is using the simulation samples for optimizing the controller to handle complex changes of contacts from highly dynamic fallings. By breaking a fall into a sequence of multiple contact, like "UKEMI" of Judo, we expect the robot to endure larger external perturbations. In addition, our simulation-based algorithm allows us to incorporate an arbitrary objective function so that we can prioritize the body parts to be protected.

The development of a safe falling controller requires design decisions on when to detect the falling and how to evaluate the damages from falling. In this proposal, we assume that the falling can be easily detected by observing acceleration of the center of mass and the falling controller will be activated after ?? seconds. Evaluating the damages from the falling might be an interesting problem to us, because it will dramatically affect the optimal control policy. We plan to measure the damages on the bodies and joints as contact forces and joint constraint forces, which might be scaled to select more important ones to be protected. Therefore, the objective function of our opitmization is accumulated body damages and joint stresses while ignoring the negligible value under the threshold.

2.2.2 Related Work

Safe falling and landing for bipeds is a topic that receives broad attention in many disciplines. Robotic researchers are interested in safe falling from standing height for the purpose of reducing damages on robots due to accidental falls. Previous work has applied machine learning techniques to predict falling [16], as well as using an abstract model to control a safe fall [10, 11, 24]. In contrast to the related work in robotics, we use simulation samples with detailed robot models to generate the optimal control policy. The main advantage of using simulation is that it can capture complex and arbitrary changes of contacts, which is hard to be formulated with an abstract model. We draw inspiration from kinesiology literature and sport practitioners. In particular, the techniques developed in freerunning and parkour community are of paramount importance for designing landing control algorithms capable of handling arbitrary scenarios [8, 3].

2.2.3 Algorithm

Optimal control for a single scenario As the simplified version of the problem, we first develop the falling controll for a single scenario, which starts from the given initial state. For instance, the robot starts from its initial standing pose and being push its head backward for 0.1 second with 10N force. When we know the parameterization of the controller, optimizing control parameters for the given scenario can be easily solved by various optimization techniques, such as Covariance Matrix Adaptation (CMA) [15]). After solving one instance of the scenarios, we can use the solution as the guidance for the rest of the scenarios.

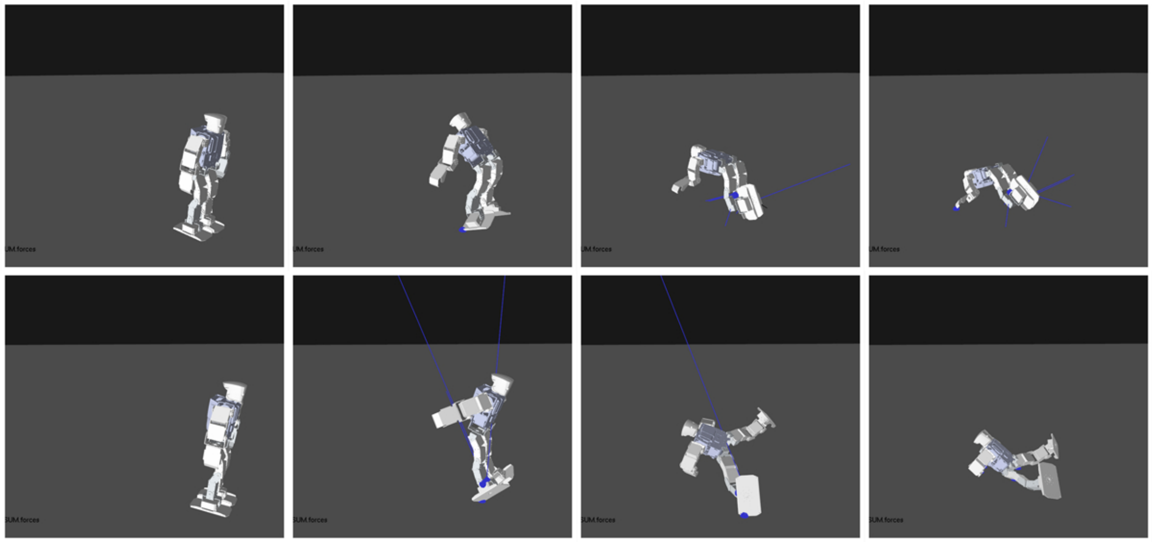

Figure 2: Optimized falling motions with linear joint trajectories (Top) and bezier trajectories (Bottom)

However, finding the proper parameterization of the controller is a very difficult problem which usually requires a lot of prior knowledge. In fact, there exist numerous control options in robotics, such as pose control, torque control, virtual force control using Jacobian Transpose, and so on. Even one of the option, a pose control has an infinite number of choices for representing its joint trajectories with splines. Indeed, the selection of control dimension has a huge impact on the result: we tested two parameterization of controller: a pose tracking with linear segments and bezier curves, which the latter has four times more degrees of freedom than the former. The optimization indicates that the bezier curve gives us much better results, which is one third of maximum impact comparing to the linear control (Figure 2).

Therefore, our short term goal is finding the proper parameterization of the controller. One intuition from the previous falling and landing project is momentum planning can be a simple and robust solution, so finding the proper momentum trajectory with an abstract model would be a promising approach. Another potential approach is incrementally finding the control parameterization. In this approach, we search over the optimal parameterization by mutating the control dimension with genetic algorithm. The value of each control dimension will be determined by solving the optimization problem with the standard technique, like CMA.

Optimal policy for multiple scenarios Even if we have an optimal motion for a single scenario, it is not sufficient for the protection of real robots. First, the state of the robot is keep changing due to its original task (i.e. locomotion or manipulation), so we cannot assume the fixed initial state. In addition, we may not know the exact information on the current situation, such as the amount of external forces, because sensors provides us only limited amount of data that currupted by noise. Therefore, the optimal motion for the single scenario is not likely to be optimal for other situations, and further cause the severe damages on the body. To overcome this issue, we need to generate the general policy which can react to the sensor data and update the falling motion of the robot.

However, generating an optimal policy is a difficult problem which requires a lot of samples. One promising approach is reinforcement learning which optimizes the best action for the current state by incorporating the long-term rewards, which is proven to be effective both in computer graphics [7] and robotics [20]. However, reinforcment learning requires us to select the proper set of state variables which is still challenging. If the number of state variables are too many, the number of state becomes exponential. In opposite case, the state variables does not well reflect the details of the full-scale simulation. One possible direction is to use an abstract model such as the inverted pendulum (IP) to represent the state in reinforcement learning, which needs the further investigation and experiments.

2.2.4 Expected Results

In this project, we try to control safe falling of virtual and real robots which minimizes the damages on the body parts and joints. As a testbed, we select a Robovie-X Standard [2] as a target robot (Figure 3). A robovie has 17 Degrees of Freedom (Head:

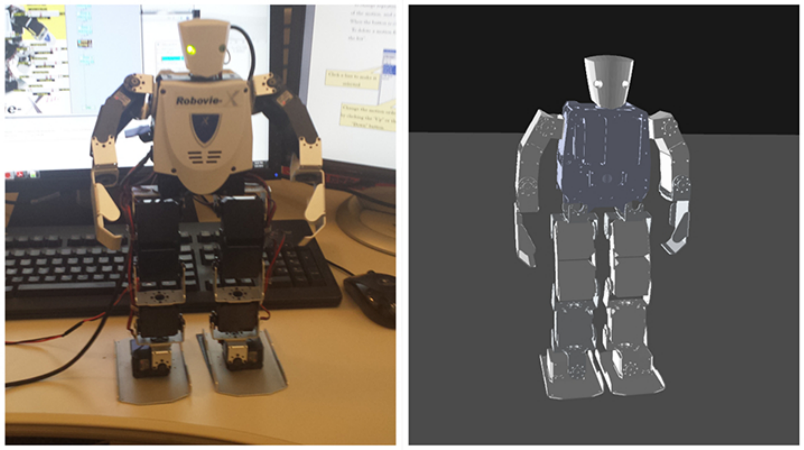

Figure 3: Real and virtual robovies (Left and Right)

1, Arm: 6, leg 10) that are operated by VS-S092J servo motors and a gyro/acceleration sensor board. The control of robovie is done by their own software, RobovieMaker, which takes keyframed trajectories as inputs. However, we have a plan to change the control framework using Arduino for more flexible control. Moreover, we prepare the virtual model of the robovie from the CAD model, which can be simulated in Dart simulator [1] (Figure 3). Then the simulation parameters such as torque limit or maximum speeds are specified from the specification of servos.

We want to generate a robust falling controller which can handle a wide range of initial states and external perturbations. It may include the different state of the robot, different environments, and different directions/strengths of pushes. For all cases, the damage should be minimized which can be verified by analyzing the simulation data or the motion captured data of robots. Further, we will experiment the objective function with user-specified constraints, such as "head should be protected" and see how the strategy will be changed.

CHAPTER III

HUMAN-GUIDED ITERATIVE TRAINING OF DYNAMIC MOTIONS

In this section, we propose iterative training frameworks for dynamic motor skills of virtual characters and real robots. We already introduced an intuitive and interactive system for developing dynamic controllers of virtual characters, inspired by human learning process [9]. Further, we want to extend the system to guide a motor skill acquition process of real robots, which is also related to "Learning from Demonstration" paradigm in robotics.

3.1 Prior Work: Iterative Training Of Dynamic Skills Inspired By Human Coaching Techniques

In our prior work [12], we introduce an intuitive and interactive framework for developing dynamic controllers inspired by how humans learn dynamic motor skills through progressive process of coaching and practices. The user only needs to provide a primitive initial controller and high-level, human-readable instructions as if she is coaching a human trainee, while the character has the ability to interpret the abstract instructions, accumulate the knowledge from the coach, and improve its skill iteratively. We introduce "control rigs" as an intermediate layer of control module to facilitate the mapping between high-level instructions and low-level control variables. Control rigs also utilize the human coach's knowledge to reduce the search space for control optimization. In addition, we develop a new sampling-based optimization method, Covariance Matrix Adaptation with Classification (CMA-C), to efficiently compute control rig parameters. Based on the observation of human ability to "learn from failure", CMA-C utilizes the failed simulation trials to approximate an infeasible region in the space of control rig parameters, resulting a faster convergence for the CMA optimization. Without using motion trajectories, or tuning any parameters, We demonstrate the design process of complex dynamic controllers using our framework, including precision jumps, turnaround jumps, monkey vaults, drop-and-rolls, and wall-backflips (Figure 4).

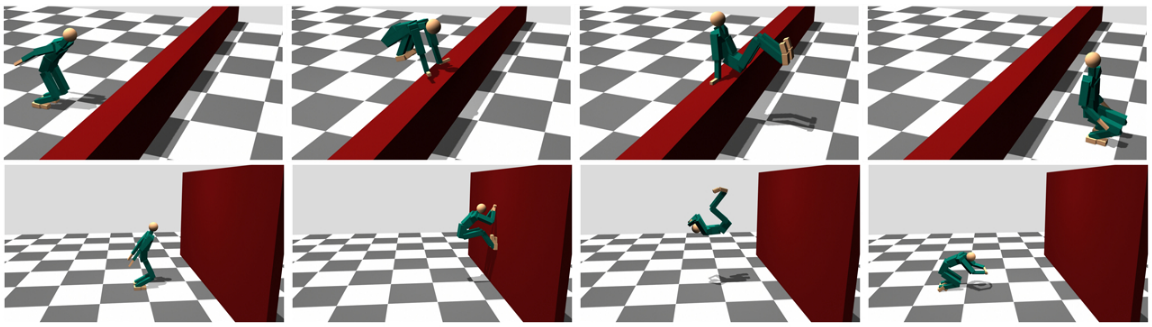

Figure 4: Monkey vault (Top) and wall-backflip (Bottom).

3.2 Learning Dynamic Skills in Control Domain for a Humanoid Robot

3.2.1 Problem Description

In this section, we propose to develop a framework for learning dynamic motor skills of humanoid robots from user-provided demonstrations and instructions, which are common ways to guide a human-trainee. Within our framework, a coach demonstrates a set of example motions that consists of joint or mementum trajectories. Since trajectories cannot be directly applied to the robots, instructions are used to interprete the motion to a proper control space, such a low-dimensional torque space or a control rig space [12] that suggested in our previous work. Finally, we derive a robust control policy from the interpreted demonstration set by learning an approximation of the state-action mapping. As a result, we can demonstrate a full-body dynamic motor skills of humanoid robots under the guidance of human coach.

3.2.2 Related Work

Learning from demonstration (LfD), also known as programming by demonstration, has been an attractive paradigm for training motor skills to robots. In this paradigm, a set of examples are provided by human teachers, and an optimal policy is generated from such examples. Since the early work of Kuniyoshi *et al.* [19], it has been proven to be effective for training motor skills in numerous task domains, including object manipulation [5, 6, 23], navigation [17], full-body motion generation [18], and so on. To increase the robustness, the learned motor skills are further generalized using machine learning techniques, such as gausian mixture model [6] or motion primitives [22]. However, dynamic motor skills of humanoids have not been fully examined yet, except the only few works on the locomotion [21], which is our target domain in this proposal.

3.2.3 Algorithms

Domain of learning Choosing the right domain of learning is a critical problem in "Learning from Demonstration" paradigm. In the literature, one of the most common domains is a set of kinematic trajectories in joint angles or task spaces. For instance, Akgun *et al.* [4] presented a framework for learning object manipulation tasks, such as scooping, pouring, or placement, from the keyframe data using Sequential Pose Distributions (SPD). However, joint or torque trajectories cannot be directly applied to the dynamic skills of the robots due to the different dynamic properties of a coach and a trainee, which can make a huge impact on the motion with just minor deviations.

To overcome this issue, we hypothesize that learning in the control domain instead of the kinematic domain would allow straight-forward learning and roboust behaviors. To this end, we combine the demonstration with user-provided high-level instructions, which can help us to identify the proper domain of controls. The control domain can be a projected low-dimensional control space using Principal Component Analysis

(PCA) or a control rig space, as defined in [12]. Especially, control rigs can project the high-dimensional control into lower dimensions by control multipel degrees of freedoms simultaneously, and can be easily easily constructed from a sequence of human-readble instructions. For instance, “MOVE DOWN” instruction will add a “Leg-distance” rig, which controls the distance between the root and feet using an inverse kinematics solver. We hope that high-level instructions can expedite the learning from demonstration for dynamic motor skills.

Optimization To apply to the trainee, a robot, the control parameters are required to be optimized to the new dynamic character to follow the user-provided examples and instructions. To optimize the parameters in the simulation environtment, it might be easily solved with a standard sampling-based optimization techniques, such as CMA. However, deploying the controller to the real robot with hardwares may require the additional robustness of the controller due to the noise on the sensors and servos. Therefore, we may need to ensure the robustness of the solution, which can be potentially done by testing the objective value with minor perturbations as suggested in Ha *et al.* [13].

3.2.4 Expected Results

CHAPTER IV

TIMELINE FOR PROPOSED RESEARCH

- 2014, Apr: present proposal to committee
- 2014, Apr - May: setup a framework for real robots
- 2014, May - Sep: working on the robot falling project
- 2014, Sep: submit the robot falling paper to ICRA 2014
- 2014, Sep - Jul: working on the robot learning project
- 2015, May - Jul: write thesis
- 2015, Jul: defense thesis
- 2015, Sep: submit the robot learning project to ICRA 2015

REFERENCES

[1] *Dart:Dynamic Animation and Robotics Toolkit.*

[2] *Robovie-X.*

[3] *How to Land a Jump in Parkour, http://www.wikihow.com/Land-a-Jump-in-Parkour*, 2011.

[4] Akgun, B., Cakmak, M., Jiang, K., and Thomaz, A. L., "Keyframe-based Learning from Demonstration Method and Evaluation," *International Journal of Social Robotics*, 2011.

[5] Atkeson, C. and Schaal, S., "Robot Learning From Demonstration," *ICML*, no. 1994, 1997.

[6] Calinon, S., Guenter, F., and Billard, A., "On learning, representing, and generalizing a task in a humanoid robot.," *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, vol. 37, no. 2, pp. 286–98, 2007.

[7] Coros, S., Beaudoin, P., and van de Panne, M., "Robust task-based control policies for physics-based characters," in *ACM Trans. Graph*, vol. 28, 2009.

[8] Edwardes, D., *The Parkour and Freerunning Handbook.* It Books, August 2009.

[9] Fitts, P. M. and Posner, M. I., "Human performance.," 1967.

[10] Fujiwara, K., Kanehiro, F., Kajita, S., Kaneko, K., Yokoi, K., and Hirukawa, H., "UKEMI: Falling motion control to minimize damage to biped humanoid robot," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 3, pp. 2521–2526, IEEE, 2002.

[11] Fujiwara, K., Kajita, S., Harada, K., Kaneko, K., Morisawa, M., Kanehiro, F., Nakaoka, S., and Hirukawa, H., "An optimal planning of falling motions of a humanoid robot," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, no. Table I, pp. 456–462, IEEE, 2007.

[12] Ha, S. and Liu, C. K., "Iterative training of dynamic skills inspired by human coaching techniques," *ACM Transactions on Graphics (Provisionally Accepted)*, vol. 33, 2014.

[13] Ha, S., McCann, J., Liu, C. K., and Popović, J., "Physics Storyboards," *Computer Graphics Forum (The proceeding of Eurographics 2013)*, 2013.

[14] Ha, S., Ye, Y., and Liu, C. K., "Falling and landing motion control for character animation," *ACM Trans. Graph*, vol. 31, no. 6, p. 155, 2012.

[15] Hansen, N. and Kern, S., "Evaluating the CMA evolution strategy on multimodal test functions," in *Parallel Problem Solving from Nature - PPSN VIII*, vol. 3242 of *LNCS*, pp. 282–291, 2004.

[16] Kalyanakrishnan, S., Goswami, A., Kim, J., KIM, Y., HURST, J., LIU, J., XUE, F., CHEN, X., KIM, M., GONG, L., and Others, "Learning to predict humanoid fall," *LEARNING*, vol. 8, no. 2, p. 245, 2011.

[17] Konidaris, G., Kuindersma, S., Grupen, R., and Barto, a., "Robot learning from demonstration by constructing skill trees," *The International Journal of Robotics Research*, vol. 31, pp. 360–375, Dec. 2011.

[18] Kulic, D., Ott, C., Lee, D., Ishikawa, J., and Nakamura, Y., "Incremental learning of full body motion primitives and their sequencing through human motion observation," *The International Journal of Robotics Research*, vol. 31, pp. 330–345, Nov. 2011.

[19] Kuniyoshi, Y., Inaba, M., and Inoue, H., "Teaching by showing: Generating robot programs by visual observation of human performance," in *Proc. of the 20th International Symp. on Industrial Robots*, pp. 119–126, 1989.

[20] Morimoto, J., Cheng, G., Atkeson, C. G., and Zeglin, G., "A simple reinforcement learning algorithm for biped walking," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 3, pp. 3030–3035, IEEE, 2004.

[21] Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., and Kawato, M., "Learning from demonstration and adaptation of biped locomotion," *Robotics and Autonomous Systems*, vol. 47, pp. 79–91, June 2004.

[22] Pastor, P., Hoffmann, H., Asfour, T., and Schaal, S., "Learning and generalization of motor skills by learning from demonstration," *2009 IEEE International Conference on Robotics and Automation*, pp. 763–768, May 2009.

[23] Ueda, J., Kondo, M., and Ogasawara, T., "The multifingered NAIST hand system for robot in-hand manipulation," *Mechanism and Machine Theory*, vol. 45, pp. 224–238, Feb. 2010.

[24] Yun, S.-k., Goswami, A., and Sakagami, Y., "Safe fall: Humanoid robot fall direction change through intelligent stepping and inertia shaping," *2009 IEEE International Conference on Robotics and Automation*, pp. 781–787, May 2009.