Generating Humanoid Robot Motions based on a Procedural Animation IK Rig Method

Pin-Chu Yang[1][2], Satoshi Funabashi[1][2], Mohammed Al-Sada[1][2][3] and Tetsuya Ogata[1][4]

Abstract— **Many researchers have explored using humanoid robots to perform various tasks within living environments. Although well-designed motions are essential for providing friendly interactions with humanoid robots, previous works have emphasized task efficiency over user-friendliness in generated motions. To bridge this gap, we propose a motion generation method for humanoid robots which is based on procedural animation with inverse-kinematics (IK) rig methods, commonly used in video games and computer graphics (CG). First, we prepared the robot's 3D model by rigging it with two different rig structures, humanoid armature and robot armature. Then we create the IK rig from the humanoid armature, and set the robot-armature's motion objectives based on the humanoid-armature transform information. Second, to make the motion adjustable, we defined a pole-vector method on the IK rig to define the middle joint's position dynamically. Therefore, using our approach, we are able to use commonly available CG character animations on humanoid robots, and adjust motions to match various contextual or task requirements. We evaluated our approach's ability to generate humanoid robots motions that mimic humanoid character's animations, as well as the adjustability of generated motions to different contextual requirements. We used three 3D CG models of humanoid robots with different body configurations. The results show that our approach is successful in generating appropriate motions on the humanoid robot rigs based on five animations. The results also show the potential of our approach to adjust motion to correspond to factors like non-planar terrain or task-specific requirements. In light of our results, we discuss the advantages of our approach and potential applications to generate interactive motions or for task requirements.**

I. INTRODUCTION

Humanoid robots are a class of robots that are designed to have a human-like form and perform various tasks at daily-living environments. In order to be effective in daily-living environments, humanoid robots must fulfill various key design characteristics, such as a friendly aesthetics, emotion-expressibility methods to convey a robot's inner states and intentions, and the ability to maintain appropriate postures and motions that comply to various contextual and task related factors.

While there exists a numerous research that focused on the overall design of humanoid robots, research that examined context-adaptive interactability and motion generation methods are scarce. Specifically, most previous research emphasized task success as the main factor for motion generation (e.g. [1], [2]). While being task-efficient is important, we believe that unnatural motions and postures would repel users from humanoid robots, yielding them unfriendly and reducing users' desire to interact with them. Therefore, an adaptive motion generation method that satisfies task-success requirements, conveys natural movements, and adaptive to different contextual factors is essential for the successful adoption of humanoid robots.

Previous works within in CG and game industry also face the same challenges in creating lively-characters with natural movements. Procedural animation is a common technique within the animation and gaming industry that relies on various methods and tools to create real-time interactive motions [3] [4] [5]. A prominent method of procedural animation is called Inverse Kinematic (IK) rig animation [6], which refers to animation creation by animating key joints-positions and orientation rather than directly on a rig (skeleton). This method can benefit animations independently from *armatures* (rigged character models) with different rig configurations.

Unlike methods that rely on playback of pre-recorded animations, IK rig can easily be used to adapt an animation to various requirements. For example, IK rig can be used to enable characters in a game to hold a flashlight and aim it at various targets. In such scenario, the player controls where the flashlight should be aimed, and the corresponding aiming-poses are calculated through IK rig animation system in real-time. These aiming poses are usually blended with other animation, such as breathing or head movement, to make a CG character looks more lively.

Although such methods are commonly used in animation and gaming industry, to the best of our knowledge, the methods that apply such techniques on humanoid robots to generate natural and task-efficient actions are scarce. Accordingly, we take a step to apply IK rig animation generation concepts to generate adjustable motions on humanoid robots. The generated motions follow the character animation and adjustment commands. Each character animation is created from motion capture (Mocap) device, and adjustment commands are mainly focused on body limbs, which are critical parts for defining poses.

In order for a humanoid robot to follow a humanoid character animation, we first prepare 2 armatures from the

---

[1] P. C. Yang, S. Funabashi, M. Al-Sada and T. Ogata are with Waseda University, Tokyo, Japan

[2] P. C. Yang, S. Funabashi and M. Al-Sada are also with Cutieroid Project: www.cutieroid.com

[3] M. Al-Sada is also with Qatar University, Doha, Qatar

[4] T. Ogata is also with WISE at Waseda University, Tokyo, Japan
kcy.komayang@akane.waseda.jp
s-funabashi@aoni.waseda.jp
alsada@dcl.cs.waseda.ac.jp
ogata@waseda.jp

Research supported by JST CREST Grant Number: JPMJCR15E3 and JST Moonshot R&D Grant Number JPMJMS2031-2-1.

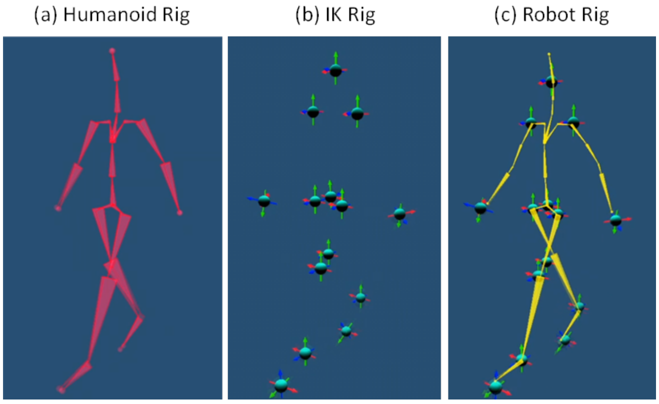


Fig. 1. The IK rig motion generation on robot starts from transferring animation (a) humanoid rig to an (b) IK rig animation. This IK rig is defined with critical joints transform information from humanoid rig, and used as objectives for IK solver to generate corresponding motion on robot rig.

same 3D model. One is with a humanoid rig, which is set up for loading animation, and another one is a robot rig that is used for actual robot action performance. An IK rig is set up from the key joints of humanoid rig as objectives for IK solver to calculate motion in real-time. The IK algorithm we used is an evolutionary algorithm based method which can create full-body, multi-objective and highly-continuous motions [7].

We focused our evaluation on two aspects: 1) Study our approach's ability to use commonly available humanoid-character animations on existing robotic rigs, and 2) to validate the flexibility of our approach to enable animation adjustments of generated animations based on various factors, like terrains, pose or task requirements. The results show our approach is generally successful in using commonly available humanoid character motions on humanoid robot rigs. Moreover, the motions can be adjusted and applied to the robot to match different or new requirements.

We discuss the results and show possible use cases that utilize motion modification on the collected humanoid character animations. Lastly, we conclude our work and discuss our future research directions.

This paper contributes with a motion generation approach based on procedural animation IK rig method for humanoid robots. Our method is based on character animation methods that are used for humanoid rigged character. Our method can reduce the required time of creating motions that are both friendly and adaptive to different factors, such as to an environment's terrain, object-manipulation needs or tasks interaction specifications.

II. RELATED WORKS

Our approach extends three strands of related research, humanoid motion translation to humanoid robots, transferring motion capture-data to humanoid robots, and procedural animation methods. We discuss each of them as follows:

Translating humanoid animation to a humanoid robot has largely been studied. For generating robot motions that are based on collected animation motions, Yamane et al [8] [9] extends the humanoid animation data to humanoid robot with whole body balance control. Choi et al [10] focused on producing natural looking motions through Bayesian optimization. Although their approach is viable to produce various motion, a major shortcoming of such approaches is that real-time motion adjustment capability was not considered. Our approach addresses this requirement by enabling the adjustment of generated motions in accordance to contextual factors that might be unforeseen in the original motion generation contexts.

Transferring real-time motion capture data to humanoid robots has been done in various previous works [11] [12] [13] [14] [15]. These studies mainly focused on remapping problems to manage the difference in structures and degrees of freedom (DoF) between human-like and robotic configurations. The methods used to capture human motion varied, where some focused on depth cameras [14] [15], while others used motion capture systems (Mocap) [11] [12] [13].

Although these approaches produce viable motions on humanoid robots, it is difficult to modify pre-recorded motions to match slightly different requirements. Some applications, such as deep learning based motion generation, consume time for data collection and intensive hardware processing resources [16]. This is crucial when it comes to manipulation tasks, such as grabbing an object with different dimensions or shapes due to a variety of object and environments.

Procedural animation refers to a set of methods for real-time animation generation instead of predefined animations (static animations) [3]. In contrary to the pre-recorded animations approaches, procedural animation is flexible enough to be adaptable to slightly different requirement. Procedural animation methods are robust for creating various types of animations, such as for four-legged animals [17], bugs-like [18], snakes, or even fantasy-creatures like dragons. In such cases, procedural animations are superior to other approaches as it is difficult or impossible to record animations using traditional methods for characters with unrealistic forms or proportions. However, to the best of our knowledge, the surveyed research is mainly limited to digital character animations without direct deployment on robots.

Our approach extends the methods presented in procedural animation for use with humanoid robots, and overcomes the disadvantages presented with previous methods that rely on pre-recorded animations. We use a Mocap system to capture input motions and generate robust and adaptive motions on a humanoid robot armature. We believe our approach has high potential to enable the advantages presented in procedural animations methods, while also being adaptable to various contextual requirements when deploying humanoid robots.

III. PROPOSED METHOD

In this section, we present the details of the proposed method of using motion generation technique with procedural animation IK rig method from a humanoid rig to a robot rig. The main concept of our approach is shown in Fig. 1. The main objective of our approach is to apply any humanoid character animation to real humanoid robots. The significance of this objective is that it would enable us to easily and directly utilize a large variety of existing character animations directly on humanoid robots. We start our method by setting up the relations between the IK rig and the robot rig. Next, in order to proceed with motion adjustment, we segregate the limbs into two main links with three joints and we treat the motion adjustments as the positional change of the end-joint, after which we define the middle-joint's position through pole-vector method. We explain the two above steps in more details as follows:

A. Setting up IK rig animation

Two rigs are needed to implement motion generation on robot based on a character animation, . *1) Humanoid Armature:* which is used for the general humanoid rig structure and it is used for loading the humanoid character animations. *2) Robot Armature:* which is a rig that represents the actual humanoid robot's body configurations (i.e. DoF, Rotation axes, links, etc...). This set up process is called as *rigging*, which is a key approach in CG animation for controlling a CG character's movements. The rigs are first created in Blender [19], after which they are imported into Unity3D [20] and configured to work with the Mecanim system, which is Unity3D's built-in animation management and configuration tool.

After importing both armatures to Unity3D, the *humanoid armature* is configured with "humanoid" setting, while the *robot armature* is configured with "generic" setting. The difference between these two settings are how the animation is controlled in different spaces; the *robot armature* uses joint space control, while the *humanoid armature* uses muscle space control [21]. A real-time full-body IK system [7] is imported and used within Unity3D, where we set up to the IK system on the robot armature with physical joint configurations.

After configuration, both armatures are placed and aligned to each other on the "hips" joint in the Unity3D scene. For mimicking full body motion, we can simply assign each IK joint-objective on the *robot armature* to the corresponding joint on the *humanoid armature*.

However, this implementation assumes equal configurations and dimensions between the armatures, therefore, applying such direct approach would results in faulty actions due to mismatch between the armatures. Accordingly, we use the *Humanoid armature* as a reference to calculate the objectives for each joint on the textitRobot armature and to fulfill adjustable motion requirement. In this study, the motion modification mainly focuses on limb motion adjustment, and the adjustment is defined with the end part of limb position change and the bend direction that described in following sub-sections.

B. Motion modification and Pole Vector

To proceed motion modification that refers to the original animation, each limb is calculated to vectors from three joints (which are $root$, mid and end) on the humanoid character. These vectors are used for adjusting changes of position of end joint and defining mid joint position for modified bending pose. Here, a $Pole$ vector is introduced to describe the limb's bending direction, which is usually included in the IK solver for game and CG industry to define bending pose of a 2-bones linked-limb. We extend this idea to define mid position of objective instead of defining pose directly since robot character's limb is usually not a 2-linked limb system but constructed with serial actuators that have over 2 links.

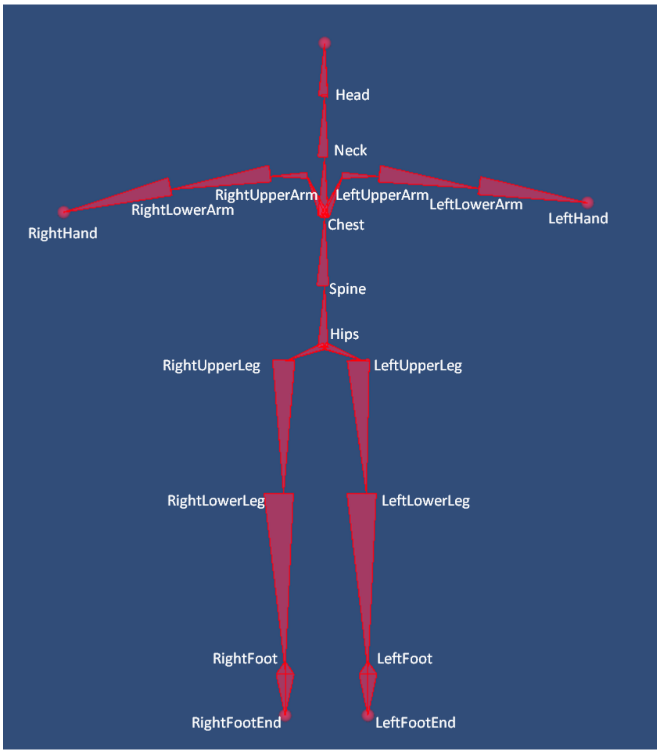


Fig. 2. The rig structure of humanoid character used for loading humanoid animation. Each joint will be used as IK solver's objectives to corresponding joints on robot rigged character.

The $Pole$ vector defined in this study is based on 2 assumptions, (1) the length of links are fixed, and (2) bending direction of modified pose has same bending direction to $root$-end link as original animation.

We first calculate the original $Pole$ vector, which is used for describing original bending pose of limbs from a humanoid character, and it is calculated by (1).

$$\overrightarrow{Pole_{orig}} = \overrightarrow{RM} - \overrightarrow{RE}(\overrightarrow{RM} \cdot \overrightarrow{RE}) \tag{1}$$

where R, E, $M \in \mathbb{R}^3$ are $root$, end and mid joint position of a limb, respectively. For example, leg's $root$ refers to *UpperLeg* joint, end refers to *Foot* joint and mid refers to *LowerLeg* joint, respectively. For ease to understand, Fig. 2 shows the rig structure with joints name, and Fig. 3 shows the idea of how the $Pole$ works to define new poses.

To calculate the new pose with an adjustment of E to E', we need to calculate the new pose's $Pole$ vector to define the new mid joint-position. This $Pole$ vector is calculated through (2), where the Rot (3) is a rotation matrix used for rotating original $Pole$ vector ($\overrightarrow{Pole_{orig}}$), and its elements

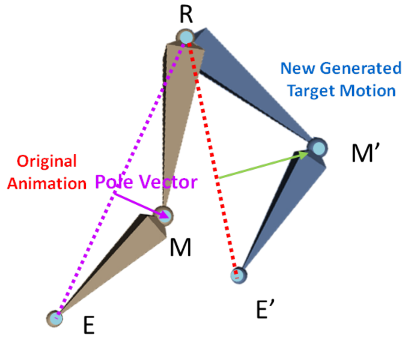


Fig. 3. The $Pole$ vector is a vector perpendicular to the RE link and points to the midpoint M. The new midpoint M' is generated with reference to the original $Pole$ vector when the new endpoint E' is adjusted.

are in (4); the unit vector of the rotation axis in (5); the trigonometric elements in (6), 7.

$$\overrightarrow{Pole} = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = Rot \cdot \begin{bmatrix} P_{x_{orig}} \\ P_{y_{orig}} \\ P_{z_{orig}} \end{bmatrix} \tag{2}$$

$$Rot = \begin{bmatrix} r11 & r12 & r13 \\ r21 & r22 & r23 \\ r31 & r32 & r33 \end{bmatrix} \tag{3}$$

$$\begin{cases} r11 & = cos\theta + a_x^2(1 - cos\theta) \\ r12 & = a_x a_y(1 - cos\theta) - u_z sin\theta \\ r13 & = a_x a_z(1 - cos\theta) + a_y sin\theta \\ r21 & = a_y a_x(1 - cos\theta) + a_z sin\theta \\ r22 & = cos\theta + a_y^2(1 - cos\theta) \\ r23 & = a_y a_z(1 - cos\theta) - a_x sin\theta \\ r31 & = a_z a_x(1 - cos\theta) - a_y sin\theta \\ r32 & = a_z a_y(1 - cos\theta) + a_x sin\theta \\ r33 & = cos\theta + a_z^2(1 - cos\theta) \end{cases} \tag{4}$$

$$a = [a_x, a_y, a_z] = \frac{\overrightarrow{RE} \times \overrightarrow{RE'}}{||\overrightarrow{RE} \times \overrightarrow{RE'}||} \tag{5}$$

$$cos\theta = \frac{\overrightarrow{RE} \cdot \overrightarrow{RE'}}{||\overrightarrow{RE}||||\overrightarrow{RE'}||} \tag{6}$$

$$sin\theta = \sqrt{1 - cos^2\theta} \tag{7}$$

With the $Pole$ vector for new bending pose is calculated ($\overrightarrow{Pole}$), a new mid joint position objective M' can be calculated through (8).

$$\overrightarrow{OM'} = \overrightarrow{OR} + \overrightarrow{RM'_{proj}} - \sqrt{1 - cos\alpha^2}||RM||\frac{\overrightarrow{Pole}}{||\overrightarrow{Pole}||} \tag{8}$$

where $O \in \mathbb{R}^3$ is the origin of the coordinate system, and M'_{proj} is the projected position of M' along the $\overrightarrow{RE'}$, α refers to the angle difference between $\overrightarrow{RM'}$ and $\overrightarrow{RM'_{proj}}$ vectors, respectively. Since the links length are fixed as defined in assumption (1), the scalar $||\overrightarrow{RM}||$ is equal to the scalar $||\overrightarrow{RM'}||$ may enable $\overrightarrow{RM'_{proj}}$ to be computed through (9) and $cos\alpha$ through (10).

$$\overrightarrow{RM'_{proj}} = \frac{||\overrightarrow{RM}||}{||\overrightarrow{RM}|| + ||\overrightarrow{ME}||}\overrightarrow{RE'} \tag{9}$$

$$cos\alpha = \frac{||\overrightarrow{RM}||}{||\overrightarrow{RM'_{proj}}||} \tag{10}$$

After preparing the limbs' objectives, the IK rig can be described as a total fitness function with objectives remapping relation between robot character and humanoid character. For mimicking motion without motion adjustment, the relation can be described as (11).

$$\mathcal{F} \in \{\mathcal{F}_{orig}(w_{orig})\} \tag{11}$$

where $\mathcal{F}_{orig}$, w_{orig} are fitness function(s) and weight(s) pairs of IK rig that are used for IK solver to update motion solution. With motion adjustment, such as motion like "extension of limbs" cases can be modified motion through adjusting objectives, as explained in the next paragraph.

Firstly, the adjustment of extension of limbs, which refers to the distance adjustment of $root$ to end joint of limb. The extension of limb is calculated with a fixed $root$ joint and adjusted end joint positions along $root$-end vector, which is defined as (12). We believe this motion modification is useful in video games, such as to create terrain adaptive animations like crouching or to let a character's hand follow an object. Such approach is also applicable for creating motions for humanoid robots, since they both share the same animation purposes.

$$\overrightarrow{OE'} = \overrightarrow{OR} + \varepsilon\overrightarrow{RE} \tag{12}$$

where ε refers to extension factor which is in a range as:

$$\frac{||\overrightarrow{RM}|| + ||\overrightarrow{ME}||cos(b)}{||\overrightarrow{RM}|| + ||\overrightarrow{ME}||} >= \varepsilon >= 1.0 \tag{13}$$

where the lower limit refers to fully bent *flexion* motion. For legs and arms, we refer to human *range of motion* study [22], in which $b_{leg} = 130\,\text{deg}$ and $b_{arm} = 145\,\text{deg}$ are the maximum flexion angle respectively.

IV. EXPERIMENT AND RESULT

A. *Experiment Design*

The main objectives of our evaluation are to 1) study our approach's ability to use commonly available humanoid character animations on existing robotic rigs, and 2) to validate the flexibility of our approach to enable animation adjustments of generated animations based on various factors, like various terrains, pose or task limitations.

Similar to previous research [23], we used three different humanoid robots models for the first evaluation, which are Types A, B and C of humanoid robots acquired from [24], [25], [26] (Fig. 4). We chose these robot models as they offer various robot configurations and commonly used within the industry and academia. The IK algorithm [7] objectives are set with position and orientation to the corresponding joints on each robot model.

B. Experiment setup

Accordingly, We focused our evaluations to study the following aspects: 1) How well our approach can generate robot motions that perform actions based on humanoid armatures on 3 different robot models 2) Validate the walking motion's adjustability in different terrains and for different levels of crouching on the Type B humanoid robot.

We captured 5 different animations (showed in the next subsection) through the inertia magnetic type of Mocap *perception neuron studio* [27]. We selected these animations as we believe they are expressive motions, and were similar to motions in previous works [10]. After the animation collection process, the raw data of the captured animation is processed and cleaned from the noise that is normally generated within collected raw data.

In our Mocap system [27], noise usually refers to magnetic distortion.

Previous works have shown it is possible to utilize deep neural networks (DNN) to successfully clean and process raw motion-capture data within optical systems [28]. Their method was especially successful in de-noising generated raw data when marker tracking is lost, within noisy positions, and marker-mislabeling issues. Accordingly, we extend their approach to de-noise our raw data by using a DNN model (Fig. 5) that was trained with massive human motion data (explained in Fig. 5).

C. Results

1) Evaluating Motion Generation: The results of the generated motion through our approach is shown in Fig. 4. The original animation performance is shown using a CG model to compare it to the motion generated on the robot rig. Every robot model has a humanoid rig that loads the animation and a robot rig for robot to perform motions. From 4, we can see that each robot follows the motions of the CG models to a great degree. The robots follow the motions and poses of the CG model with some differences that depended on each robot's body configurations.

2) Evaluating Motion Adjustability: Motion adjustability is crucial for creating interactive motions based on different requirements. Fig. 6 shows adjustment of the "walking" motion to "crouch walking" motion.

Moreover, Fig. 7 demonstrates the robot performing "walking" motion with non-planar terrain. In the figure, Type B humanoid robot's left leg are adjusted to walk on the box, while the right leg applies the animation normally. Therefore, we believe our approach has adjusted the walking animation successfully.

3) Comparing robot rig and Humanoid rig trajectory: We additionally compare the trajectories of the robot armature and the humanoid armature to see if the IK rig can properly generate motion that follows the trajectory of the humanoid armature. Fig. 8 shows Type C humanoid robot performing a "Surprise" motion. We can clearly see that the trajectory of the robot armature is following the humanoid armature.

Overall, we believe the results are encouraging to pursue further work. The results show that our approach enabled us to generate robot motions based on commonly available animations, and enabled us to adjust generated motions. The generated motions of the robot generally followed the CG character's motions, despite differences that were caused by varied body configurations between the CG character and robot models.

V. DISCUSSION

A. Extending motions to different motions

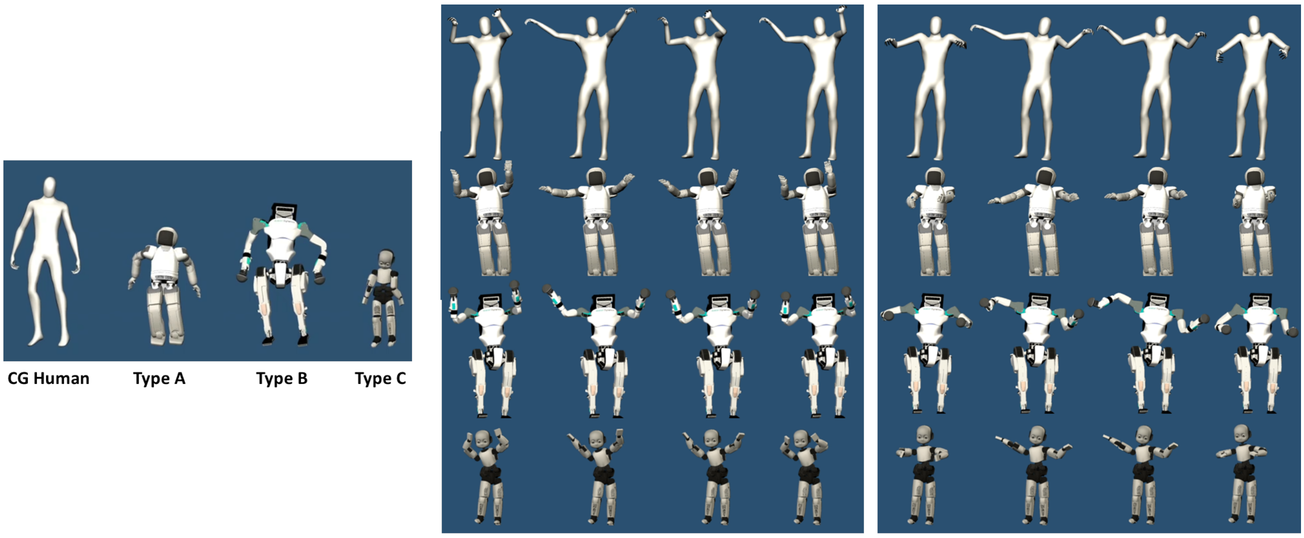
Using a procedural animation IK rig method to generate robot motions is advantageous to creating interactive and adaptive humanoid robot motions. One of the advantage for applying our approach is that the generated motions are easy adaptable to create new motions with different requirements by reassigning part of IK rig objectives. Fig. 9 shows an example of extending "walking" motion to a "carry camera and walk" motion on the Type A humanoid robot. Furthermore, by assigning additional "look at" objective from the lens of camera to the target, we can extend the "carry camera and walk" motions with "camera aiming" action as shown in Fig. 10.

B. Task space motion performance for task operation

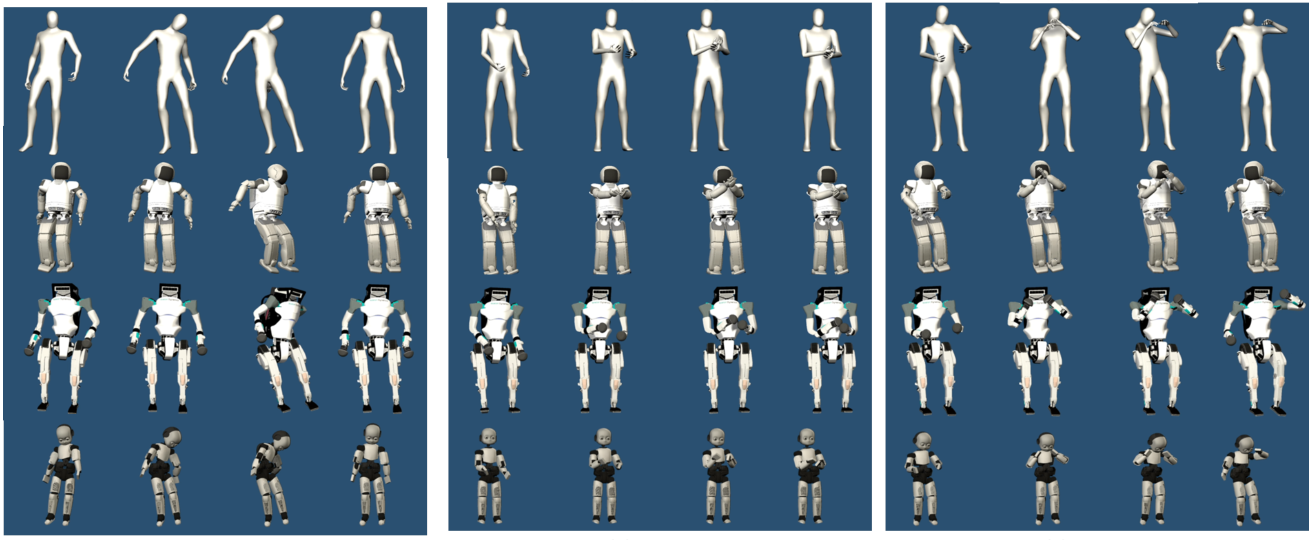
Another advantage of our approach is the independency of the motion generation process from the robots configuration, thereby enabling us to exchange the robot models or the motion's objectives. Fig. 11 shows a demonstration of transferring a CG character's "stirring" motion to Type A humanoid robot. Since the trajectory from CG character is available in robot's working range, not only the natural looking animation is conveyed, but also a precise motion can be achieved for task operation. This is an essential factor for creating natural and user-friendly motions that are also task-efficient in humanoid robots.

VI. CONCLUSION AND FUTURE WORK

In this study, we present a novel humanoid robot motion generation technique that enables utilizing pre-existing character animations from CG world on real robot rigs with motion adjustability. Our approach is based on procedural animation IK rig method to generate humanoid robot motions. A key advantage of our method is that it enables using commonly available humanoid character motions on humanoid robots with the ability to adjust motions to meet different requirements. We demonstrated the robustness of our approach with an evaluation on three different humanoid robot models. The results show that the humanoid robots could achieve various adaptive-motions based on collected humanoid character animations. The results not only show that we are able to animate a robot in a natural and friendly fashion by effectively using CG animation technology, but also enables making new motions from already collected animations by adapting them to various environments or



(a) Greeting (b) Surprise



(c) Look Around (d) Cross Arm (e) Crying

Fig. 4. Five different motions performed by a CG human, Type A, Type B, and Type C humanoid robot from the top to the bottom. These robots with different body configurations follow the motions performed on humanoid rig by assigning IK rig objectives to corresponding joints.

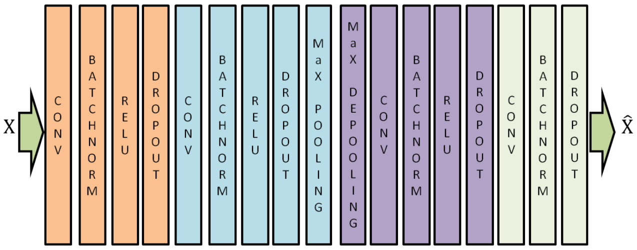


Fig. 5. The structure of the de-noising DNN model. $\mathbf{X}$ stands for a input T length long of motion sequence data. The output $\hat{\mathbf{X}}$ stands for the de-noised motion sequence. We trained this model with Hdm05 dataset [29], CMU [30], MHAD dataset [31] and the dataset is used in [32].

task specifications. Furthermore, with extended configuration of the rig's objectives, manipulation tasks are also possible and easy to apply from different body configurations (As discussed in Section V).

Our future work is focused on two main directions. First, we intend to extend our method to take into consideration other important factors in humanoids robots, such as mimicking motion with balancing or context-awareness in real world deployment environments [33]. Our current approach focuses on generating motions and applying them within CG simulation. However, applying the generated motions to real humanoid robot encompasses numerous challenges, including mechanical design constraints and physical motion attributes like acceleration, limb weight or inertia. Therefore, such constraints and attributes must be considered when applying the final motions to different humanoid robots. Second, we would like to deeply evaluate our approach by implementing and studying it on a real humanoid robot within various contexts, such as for entertainment or for physical manipulation by combining with imitation learning method [34] [35]. A realistic deployment of our approach on a real humanoid robot would enable us to further deeply

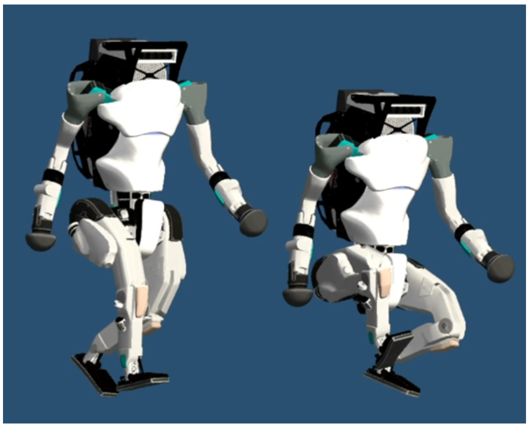


Fig. 6. The crouching walking motion (right) performing on Type B humanoid robot is created from walking motion by adjusting extension of leg's IK rig that can provide high variety of motions for robot to perform terrain adaptive actions.

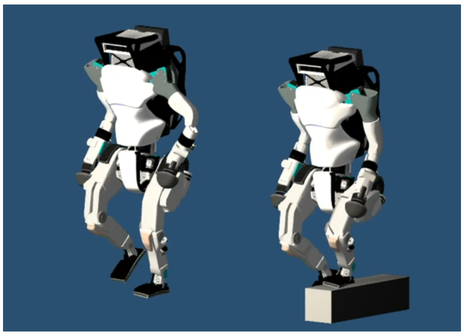


Fig. 7. The leg extension adjustment method can be used to control the robot to generate motion generation for uneven grounds. Left side shows a robot performing "Walking" motion, and the right side shows an adaptive walking motion.

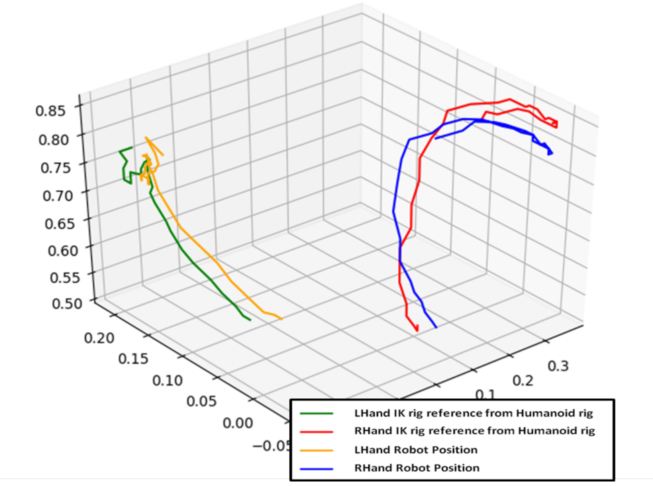


Fig. 8. Both hands position target trajectory and actual operated trajectory of Type C humanoid rig and robot rig to perform "Surprise" motion. We can see that the robot rig moves in accordance with the trajectory of the humanoid rig.

explore the challenges of utilizing our approach within real world contexts.

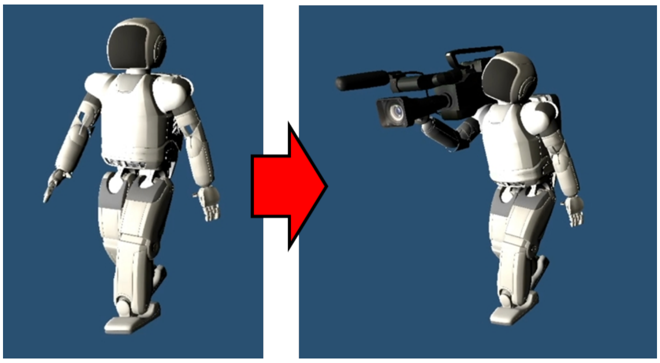


Fig. 9. The "camera carrying walking" motion is created from "walking" motion by mounting the camera on the right shoulder and assigning the camera grip as a target for right hand.

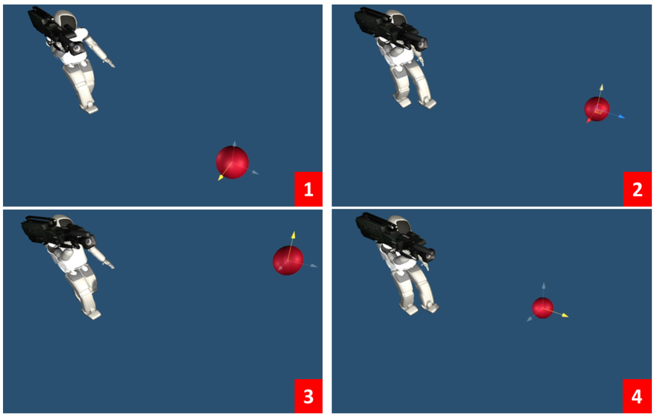


Fig. 10. The interactive aiming motion while carrying camera performed by a robot. In this figure, we move the red sphere, that represents the aiming position for robot, to perform a corresponding aiming motion.

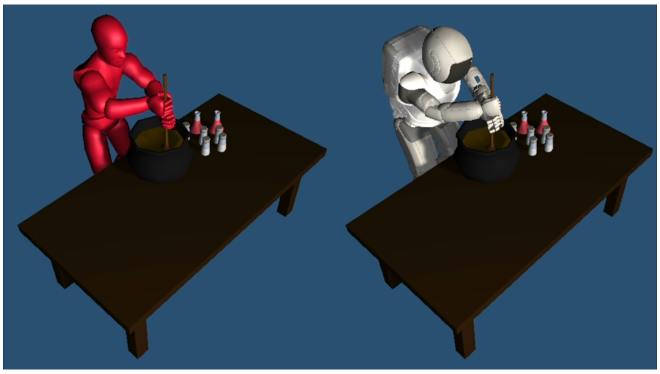


Fig. 11. By assigning positional and orientation objectives to corresponding joints, the stirring task from CG character can be performed on Type A humanoid robot with sufficient precision.

ACKNOWLEDGEMENTS

This work was supported by JST CREST Grant Number: JPMJCR15E3 and JST Moonshot R&D Grant Number JPMJMS2031-2-1.

REFERENCES

[1] B. Tang, T. Chen, and C. G. Atkeson, "Humanoid full-body manipulation planning with multiple initial guesses and key postures," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 6395–6401.

[2] D. Leidner, W. Bejjani, A. Albu-Schaeffer, and M. Beetz, “Robotic agents representing, reasoning, and executing wiping tasks for daily household chores,” in *Proceedings of the 2016 International Conference on Autonomous Agents*, ser. AAMAS ’16. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2016, p. 1006–1014.

[3] G. Deglorie, K. Samyn, P. Lambert, R. Walle, and S. V. Hoecke, “Procedural animation of human interaction using inverse kinematics and fuzzy logic,” in *GRAPP*, 2015.

[4] Z. Bhatti, I. A. Ismaili, S. Zardari, H. A. M. Malik, and M. Karbasi, “Procedural animation of 3d humanoid characters using trigonometric expressions,” 2016.

[5] H. V. Welbergen, J. Zwiers, and Z. Ruttkay, “Real-time animation using a mix of physical simulation and kinematics,” *Journal of Graphics, GPU, and Game Tools*, vol. 14, pp. 1 – 21, 2009.

[6] A. Bereznyak, “IK Rig: Procedural Pose Animation,” in *2016 Game Developers Conference Talk*, 2016, [Accessed: 2021-08-06]. [Online]. Available: https://www.gdcvault.com/play/1022984/IK-Rig-Procedural-Pose

[7] S. Starke, N. Hendrich, D. Krupke, and J. Zhang, “Evolutionary multi-objective inverse kinematics on highly articulated and humanoid robots,” *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-Septe, no. September, pp. 6959–6966, 2017.

[8] K. Yamane and J. Hodgins, “Simultaneous tracking and balancing of humanoid robots for imitating human motion capture data,” *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2510–2517, 2009.

[9] K. Yamane, S. O. Anderson, and J. K. Hodgins, “Controlling humanoid robots with human motion data: Experimental validation,” in *2010 10th IEEE-RAS International Conference on Humanoid Robots*, 2010, pp. 504–510.

[10] S. Choi and J. Kim, “Towards a Natural Motion Generator: A Pipeline to Control a Humanoid based on Motion Data,” *IEEE International Conference on Intelligent Robots and Systems*, pp. 4373–4380, 2019.

[11] F. J. Montecillo-Puente, M. N. Sreenivasa, and J. P. Laumond, “On real-time whole-body human to humanoid motion transfer,” *ICINCO 2010 - Proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics*, vol. 2, no. January, pp. 22–31, 2010.

[12] S. Kim, C. H. Kim, B. You, and S. Oh, “Stable whole-body motion generation for humanoid robots to imitate human motions,” *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, pp. 2518–2524, 2009.

[13] Y. Ishiguro, K. Kojima, F. Sugai, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, “High speed whole body dynamic motion experiment with real time master-slave humanoid robot system,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 5835–5841, 2018.

[14] I. Almetwally and M. Mallem, “Real-time tele-operation and tele-walking of humanoid robot nao using kinect depth camera,” in *2013 10th IEEE INTERNATIONAL CONFERENCE ON NETWORKING, SENSING AND CONTROL (ICNSC)*, 2013, pp. 463–466.

[15] J. H. Sohn, S. Oh, C.-H. Lee, and S.-S. Kim, “Recursive inverse kinematic analysis for humanoid robot based on depth camera data,” *Journal of Mechanical Science and Technology*, pp. 1–11, 2021.

[16] J. Pankert, L. Kaul, and T. Asfour, “Learning efficient omni-directional capture stepping for humanoid robots from human motion and simulation data,” in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, 2018, pp. 431–434.

[17] Z. Bhatti, A. Shah, and F. Shahidi, “Procedural model of horse simulation,” *Proceedings - VRCAI 2013: 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, no. November, pp. 139–146, 2013.

[18] A. A. Karim, T. Gaudin, A. Meyer, A. Buendia, and S. Bouakaz, “Procedural locomotion of multilegged characters in dynamic environments,” *Computer Animation and Virtual Worlds*, vol. 24, no. 1, pp. 3–15, 2013.

[19] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018, [Accessed: 2021-06-28]. [Online]. Available: http://www.blender.org

[20] Unity Technologies, “Unity real-time development platform 3d, 2d vr & ar engine,” https://unity.com/, 2021, [Accessed: 2021-05-06]. [Online]. Available: https://unity.com/

[21] A. L. C. Ruiz, C. Pontonnier, N. Pronost, and G. Dumont, “Muscle-based control for character animation,” *Computer Graphics Forum*, vol. 36, 2017.

[22] K. Yonemoto, S. Ishigami, and T. Kondou, “関節可動域表示ならびに測定法[Range of motion and measurement method],” リハビリテ-ション医学*[The Japanese Journal of Rehabilitation Medicine]*, vol. 32, no. 4, pp. p207–217, apr 1995. [Online]. Available: https://ci.nii.ac.jp/naid/40003744193/

[23] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, “Deepmimic: Example-guided deep reinforcement learning of physics-based character skills,” *ACM Trans. Graph.*, vol. 37, no. 4, July 2018. [Online]. Available: https://doi.org/10.1145/3197517.3201311

[24] 3d_molier, *Asimo Robot Honda Riggedby 3d_molier*, 2010, [Accessed: 2021-07-06]. [Online]. Available: https://www.turbosquid.com/3d-models/3d-asimo-robot-honda-rigged/510978

[25] BaronDaler, *Atlas Robot Boston Dynamics Low-poly 3D model*, 2021, [Accessed: 2021-07-06]. [Online]. Available: https://www.turbosquid.com/3d-models/3d-model-atlas-robot-boston-dynamics-1442483

[26] Istituto Italiano di Tecnologia, *C4DPlugins*, 2018, [Accessed: 2021-07-25]. [Online]. Available: https://github.com/robotology/c4d-icub-plugins

[27] Noitom, “PERCEPTION NEURON STUDIO SYSTEM,” https://neuronmocap.com/perception-neuron-studio-system, 2021, [Accessed: 2021-05-04].

[28] D. Holden, “Robust solving of optical motion capture data by denoising,” *ACM Trans. Graph.*, vol. 37, no. 4, July 2018. [Online]. Available: https://doi.org/10.1145/3197517.3201302

[29] M. Meinard, M. Clausen, B. Eberhardt, A. Weber, M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber, “Documentation Mocap Database HDM05,” *Technical Report*, vol. CG-2007-2, 2007. [Online]. Available: http://cg.cs.uni-bonn.de/en/publications/paper-details/cg-2007-2/

[30] Carnegie Mellon University, “Carnegie-Mellon Mocap Database,” http://mocap.cs.cmu.edu/, 2021, [Accessed: 2021-02-29].

[31] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, “Berkeley mhad: A comprehensive multimodal human action database,” in *2013 IEEE Workshop on Applications of Computer Vision (WACV)*, 2013, pp. 53–60.

[32] D. Holden, J. Saito, T. Komura, and T. Joyce, “Learning motion manifolds with convolutional autoencoders,” *SIGGRAPH Asia 2015 Technical Briefs, SA 2015*, pp. 1–4, 2015.

[33] P. C. Yang, K. Suzuki, C. C. Chiu, T. T. Pradhono, N. Yalta, K. Kuo, K. H. Shu, and T. Ogata, “Effective imitation learning robot platform using game engine: Autonomous humanoid figure “hatsuki” mk.i,” *JSME CONFERENCE ON ROBOTICS AND MECHATRONICS*, vol. 2020, pp. 2A1–D04, 2020.

[34] P. C. Yang, M. Al-Sada, C. C. Chiu, K. Kuo, T. P. Tomo, K. Suzuki, N. Yalta, K. H. Shu, and T. Ogata, “Hatsuki : An anime character like robot figure platform with anime-style expressions and imitation learning based action generation,” in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2020, pp. 384–391.

[35] P.-C. Yang, K. Sasaki, K. Suzuki, K. Kase, S. Sugano, and T. Ogata, “Repeatable folding task by humanoid robot worker using deep learning,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 397–403, 2017.