Animation Generation of Injured Gait based on Body Status using Phase-Functioned Neural Network

Quentin Ferre
IMAC - ESIPE
Paris-Est Marne-la-Vallée University
Paris, France
quentin.ferre@outlook.fr

Tomohiro Mori
Graduate School of Systems Information Science
Future University Hakodate
Hakodate, Japan
g2117047@fun.ac.jp

Kaoru Sumi
Graduate School of Systems Information Science
Future University Hakodate
Hakodate, Japan
kaoru.sumi@acm.org

***Abstract*— We developed a system that generates character animations that consider several factors including the state of the character and the conditions of its surrounding open-world gaming environment, using a phase-functioned neural network (PFNN). The PFNN is a new neural network architecture that computes weights for every frame using a phase input. When used for character control, it enables the character to adapt its behaviors to various environments and user inputs, producing superior quality animations compared to existing methods. This study presents an approach that adapts character animations to injuries and accounts for the severity of the injuries to represent character injuries using the existing PFNN system. To evaluate the proposed system, we tested scenarios involving single and concurrent injuries to various body parts with different severity levels while the character walked or ran. Using the proposed approach, the character adapted realistically to various scenarios, accurately representing the effects of its injuries, environment, and user inputs, while maintaining a fast prediction time.**

Keywords—animation generation, body status, injured gait, phase-functioned neural network

I. INTRODUCTION

Recently, many researchers have begun to explore new ways to produce animations. The increasing popularity of this topic is a response to the ever-evolving needs of video game producers. Demand is increasingly acute owing to the recent trend of open-world gaming, in which the quantity of animation required for different characters is directly related to the scale and variety of the game environment. In these games, the player is granted a high degree of autonomy in terms of exploration and interactions. This extended range of input variables dramatically increases the workload during the development phase, especially for the animation team, who must capture, edit, and implement a massive range of animation data into the game.

The above situation explains the growing interest in alternative methods to generate character animation. As described in Section II, some methods use reinforcement learning, rewarding a character's progress through the environment to encourage the agents to learn how to move properly. Other methods, such as procedural animation, attempt to define a set of rules that will generate the appropriate motion for any possible situation. In contrast to the preceding two methods, which explore non-data-driven solutions, other methods, such as the proposed method, attempt to reduce dependence on animation data to produce motion. The phase-functioned neural network (PFNN), upon which our work is based, adopts the following approach: The system is trained using animation data, such as motion capture data. Using these training data, the system learns how to produce realistic motion by applying changes to different joints of the character body, based on the environment, player inputs, and the current character state. Once trained, the system should be able to adapt the character's animations to new situations.

In the original system [13], the character can walk, run, jump over, and crunch to pass various obstacles. This research explores a new feature based on the original system. The new feature introduces the capacity for the character to be injured in various ways and then responds by realistically adapting the animation.

II. RELATED WORK

There are many approaches to animating a three-dimensional (3D) model. In the video game industry, the most commonly used methods are handmade animation and extraction of the animation clip from motion capture data. The rapidly increasingly complexity of 3D environments and character behavior is reflected in exponential growth in demand for various animations. Fulfilling this need can be expensive, in terms of money, time, and other resources. Many of these high cost resources are related to the requirements for qualified staff and proper motion capture equipment. Thus, interest in alternative methods that require little to no human intervention is growing.

Procedural animation is used to generate character motion. Instead of using animation data, procedural animation constructs a set of rules and behaviors informed by biological

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

systems. These parameters then direct simulations for physics-based characters. This physics-based animation approach derives animations from the results of physics simulations [1], [2]. Many studies have investigated various enhancements, extensions, and improvements to procedural animation. One system considered friction [3] and another system was created as an object-based animation editor [4]. Another system generated and tuned walking and running motions in rigid-body simulations [5]. Some systems explore techniques to strengthen the human resemblance of character bodies, as did Thomas et al. in their character control system based on muscle simulations [6]. There is a system that generates injury animation using procedural animation [7].

Procedural animation methods with physics-based simulations are most suitable for "passive actors" in games, such as clothing, hair, or water. Kinematic animation methods are preferred for "active actors" such as the character body. Because procedural animation systems require powerful computers to function properly, they are unable to spontaneously adapt to untrained situations. Whereas, kinetic animation systems do not require pre-existing animation data, which enables them to realistically adapt to unanticipated events. Neural networks offer many promising applications in digital games. They can be used to accomplish many tasks including animation generation [8]. One study used a neural network to develop a system that animated the hand of a violinist [9].

Recently, animation generation systems that use reinforcement learning have emerged as a trend. The objective of these systems is to promote the development of locomotion skills by 3D characters so that they generate their own animation [10]. In this method, the character, referred to as the "agent," receives information from its environment and applies a reward function to maximize the reward earned by correct actions.

Although a simple reward function may theoretically be capable of enabling the agent to learn complex behaviors, complex reward functions are more commonly applied as results improve as the complexity of the reward function increases. However, the more complex a reward function is, the more tightly constrained it will be to specific environments or applications. Using a deep Q-network (DQN), [11] demonstrated that the emergence of rich and robust behaviors depends on the environment in which the agent is trained. Other new systems have been developed to generate and edit animation, such as the convolutional neural network (CNN) developed by Holden, Saito, and Komura [12]. Although these methods show potential, the realism of the animation that they produce still does not match that of classical kinematic animation methods, likely because the results are highly dependent on the heuristic reward function applied and the agent's training environment.

III. Animation Generation System

In this section, we first describe the existing PFNN system upon which we built our approach. Next, we describe our injury representation and explain how this information can be extracted from different animation data. Finally, we detail the changes made to the original system to implement our new feature.

A. *Phase-Functioned Neural Network*

Initially proposed in 2017, the PFNN [13] is a new architecture of neural network. In contrast to traditional neural networks, the PFNN uses a new type of input, the phase, which corresponds to the motion cycle's timing of the character's animation. The phase is used to compute the neuron and connection weights of the neural network (NN) in a cyclical function, referred to as the phase function.

The phase value depends directly on the animation cycle, as depicted in Figure 1. The frame in which the character's right foot touches the ground is labeled with a phase value of 0. The next time the left foot touches the ground has a phase value of π. The frame of the next right foot contact has a phase value of 2π. Classic interpolation is then executed to label the remaining frames. The PFNN proposed by [13] analyzes the phase, user-controlled parameters, the state of the character in the previous frame, and the environmental parameters of frame *x*. The proposed system then applies its analysis to predict the change in phase, the state of the character in the current frame, the movement of the root transform, as well as the character's trajectory.

The PFNN developed by Holden et al. [13] produces high quality motion animations such as walking, running, climbing, and crouching, even in complex environments, and all while remaining highly reactive to user interactions. Thus, the PFNN has numerous potential applications as a method of character control in virtual environments such as video games.

B. *Injury Feature*

Our goal is to represent an injury in a way that is simultaneously human-comprehensible and an efficiently processed NN input.

We took inspiration from the pain scale model, which is used by doctors to measure the intensity of a patient's pain. The Numerical Pain Rating Scale (NPRS) [14], [15], is a scale that ranges from zero to ten, where zero represents "no pain" and ten expresses the "worst possible pain." Often used by patients with communication difficulties, such as children and the impaired, the NPRS enables more accurate evaluation and diagnosis, thus improving treatment efficacy and patient outcomes.

Our system represents injury by assigning an injury level value to each of the links between the different joints in the character's body, as observed in Figure 2. This injury value is manually assigned by applying the parameters and principles of the NPRS to how the character animation conveys the character's pain in a specific part of its body. Thus, status information for every link in every frame of the animation was incorporated as additional input in our NN, as detailed in Table I.

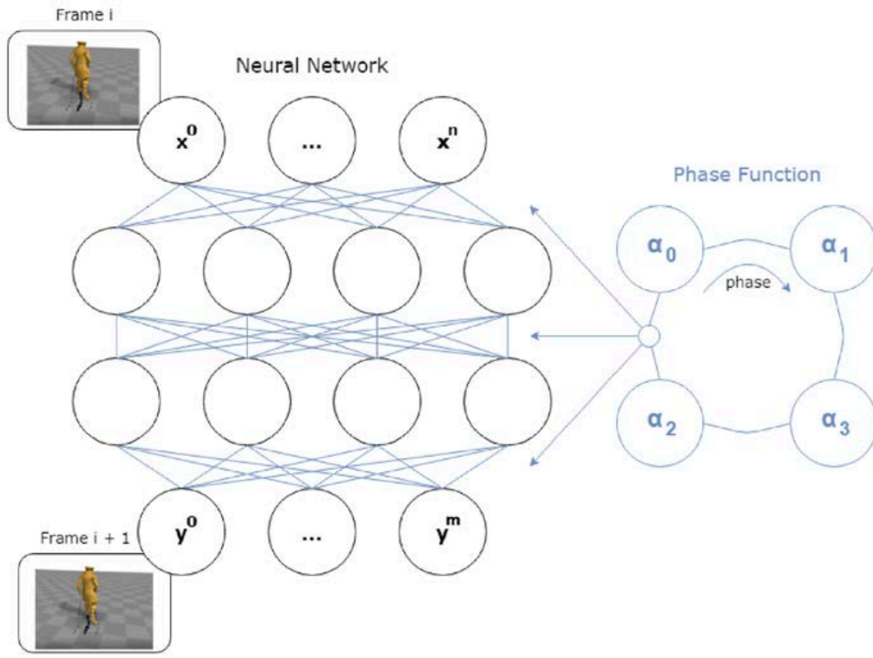

Fig. 1. PFNN visual diagram. The weights between each unit are computed for each frame using the phase function.

Next, we developed a status file generator for sets of links associated with individual joints, which, in composite, produced a file representing the whole-body status of the character in every frame.

We used the following format:

frames:x

1-i Joint_Name_1-Joint_Name_2[Injury_severity_value_1];
i-x Joint_Name_1-Joint_Name_2[Injury_severity_value_3];

ex1:
frames:225
1-225 RightUpLeg-RightLeg[10];

ex2:
frames:2000
1-500 LeftShoulder-LeftArm[7];

500-2000 LeftShoulder-LeftArm[5];LeftUpLeg-LeftLeg[10];

where *ex1* corresponds to the injury file linked to an animation in which the character is injured between its right upper leg joint and its right leg joint with a severity level of ten throughout all 225 frames of the animation, and *ex2* corresponds to an animation in which the character is injured between its left shoulder and its left arm with a severity level of seven in the first 500 frames of the animation. The rest of the animation corresponds to injuries in two different parts of the character's body with different severity levels. Using these types of data, we can quickly generate the associated matrix that will function as input for the NN.

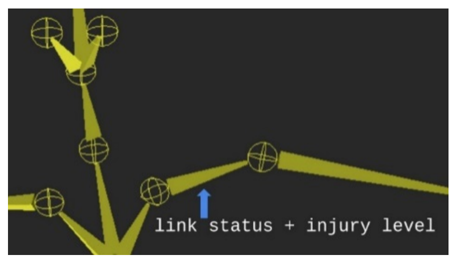

Fig. 2. Link between two joints

TABLE I.
LIST OF THE CHARACTER'S JOINTS (L/R:LEFT AND RIGHT)

Hips	L/R hip joint	L/R upper leg	L/R leg
L/R foot	L/R toe base	Lower back	Spine
Spine1	Neck	Neck1	Head
L/R shoulder	L/R arm	L/R forearm	L/R hand
L/R finger base	L/R hand index	L/R thumb	

C. Data

We collected injury animation data from two websites[1,2]. However, most of the data were too short to extract the phase value. To address this limitation, we developed a program that extends animations to eight loops. To extend the animation, we first copy the Biovision Hierarchy (BVH) animation data for every frame of the animation. We then repeated them a certain

[1] https://www.mixamo.com/
[2] http://mocap.cs.cmu.edu/

number of times, while adding a delta on the *z*-position of every joint. This means that we update the position of every joint along its forward direction, by adding the value of the same joint's *z*-position at the last frame of the previous loop cycle. This method was developed for the animation files we collected, and thus the method, like the files only moves a character forward.

Additionally, we developed two short programs that can extract the foot-to-ground contact time and the phase value of each frame from animation data, based on the methods described in Holden et al.'s original PFNN research paper [13]. Each program extracts its respective data to a text file. For this study, the program tracking foot-to-ground contact set the instance of left foot-to-ground contact as 0.0 and the subsequent timing of the right foot-to-ground contact as 0.5 and interpolated the values of the frames therebetween.

D. System

This section describes the major modifications that we made to the original PFNN system, most of which concern input parameterization. Neural network (NN) inputs are processed by an NN system to produce a certain predictive output. In this case, the input corresponded to the character's environment, the user controller input, and information about the character's animation. More specifically, the original parameterization of the input control for a single frame consisted of a vector containing the past and future positions and the directions of the character trajectory, the trajectory gait, the current position and velocities for every joint, as well as information about the environment. To incorporate our injury feature in the PFNN system, we modified the components related to input parameterization in the database, the training phase, and the demo code. It was not necessary to modify the NN structure or the output parameterization to achieve the goal of this research.

First, we modified the generation of the original database. In this step, data from various files were extracted and used to create a database file that provided access to all the PFNN input data for every frame of every animation. To incorporate the new input, we modified the database generation process so that every frame could access the status, that is, the injury severity, of all 30 of our joint links. We then modified the system's training code so that it correctly loaded the new database elements and applied the new data as inputs for the PFNN. With these changes, we thus added 30 input elements to the input layer in the form of numerical values corresponding to the new database elements. The intermediate layer was unchanged from that of the original system. Finally, we modified the demo code to consider the new inputs in calculating body condition. We mapped each joint link to a key on the keyboard, so that pressing the key increased the associated joint's injury level. We further modified the code to include these additional elements as inputs.

IV. Experiment

The experiment evaluated the behavior of the proposed system using six animations of walking characters with various injuries to their arms, legs, and feet, two animations of injured characters running, and four animations of uninjured walking and running locomotion. The experiment was executed using an AMD Ryzen 7 1700X processor (3.40 GHz) and a GTX 1060 (6 GB) using the Spyder IDE. As noted in Table II, the training phase took longer for the proposed model than in the original PFNN [13] .

We first focused our experimentation on the evolution of one joint's link injury severity. When the results of left and right links were equal, we expressed them as a single value. We observed that as the injury severity of a link increased, the animation followed a gradient from the uninjured animation, with a severity level of zero, to the fully injured animation, with a severity level of ten. This gradient governed the animation itself and the overall speed at which the character moved. With severity levels between zero and three, the character animation only differed slightly from the default, uninjured animation. When the severity was between four and six, character speed changed and the injured animation began to resemble the corresponding training animation data. Finally, at severity levels from seven to ten, the animation and character's speed became identical to those of the training data.

TABLE II.
EMORY SIZE AND TRAINING TIME FOR THE TWO SETS OF DATA TESTED

Experiment	Animation files	Database	Training time
Normal experiment	***11(40 Mb)***	***248 Mb***	***~50hrs***
Light experiment	9(26 Mb)[a]	165 Mb	~30hrs

We observed that our system produces consistent results and does not switch animation, even if it was not trained on the injured body part. Unfortunately, when mixing uninjured and injured walking animations, the system maintained the maximum speed. Additionally, with upper body injuries, one arm seemed to exhibit rotation errors. Because the error only occurred in one joint in a single animation, this may have been caused by the quality of the animations used to train the system. This theory was substantiated when this animation was removed from the training set and all these errors disappeared, as shown in Figure 3.

We next tested system performance when processing multiple simultaneous injuries. We started with an injury with a fixed severity level of five. Then the injury severity level of a second injury was increased to level ten. We retested the first case after increasing the injury severity level of the first injury to ten. In the first case, the two injured animations were blended until the second injury severity level reached a value of five. Until this point, the character animation corresponded to the animation of the most severely injured body part and progressed up the scale to a severity of ten, as observed in the first experiment. In the second case, the character's animation started to reflect a combination of the two injured animations when the second injury reached a level of three. As the second injury's severity level increased, the animation incorporated greater influence from the blended animations of the two injury animations. When the second injury reached a level of eight, the depth and influence of the injury blend was "capped." From this point, only character speed evolved. When mixing injuries, we noticed that at equal injury severity level values, character speed corresponded to the injury animation with the highest speed, as observed in Figure 4.

We also tested the proposed system on a single injury, to focus on the transition from a walking state to a running state. In the first trial, using a body part trained only with injured walking animations, the character's animation corresponded to a mix of injured walking and uninjured running animation. As before, up to a severity of three, only slight differences are apparent in the uninjured running animation. Between level four and level seven, the animation more closely resembles the injured animation, while the character speed is close to that of the uninjured running animation. Starting at an injury severity level of eight, character speed decreases, but does not slow to walking speed. At this point, if a body was trained with both walking and running injured animations, the running animation seemed to overwrite the walking animation, such that both character speed and animation while walking and running became similar regardless of severity level.

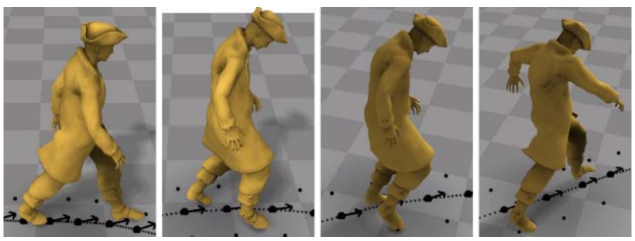

Fig. 3. Gimp walking by right leg – Right Foot joint with injury severity, from left to right, 0, 4, 7 and 10

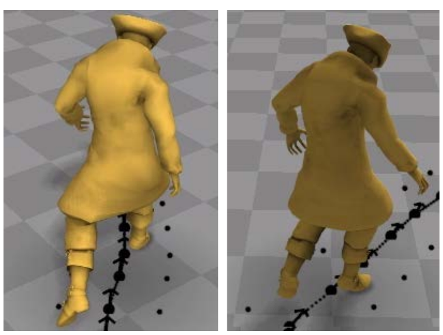

Fig. 4. Gimp walking by left leg – Left Foot joint with constant injury level of 5. Right Leg – Right Foot joint with, from left to right, injury level of 0 and 5.

Finally, we divided the value of each injury severity to reduce their influence as inputs in the overall system. This appeared to smooth the character animation and the transitions from one animation to another because the injured animation combined more fluidly with the uninjured animation. However, the visual and conceptual impact of the injured animation was reduced by this smoothing process. In terms of performance, our system achieved an efficient prediction time given the number of added inputs, as delineated in Table XVII.

To test the overall quality of our resulting animations, we conducted an experiment involving ten subjects, students from the Future University of Hakodate, nine men and one woman, ranging in age from 22 to 24 years old. The subjects viewed a series of thirteen videos corresponding to the different experiments previously described. Four of the videos focused on a single body part, five of the videos depicted multiple body parts being injured, and the last four videos observed changes in gait. We noted that question five had the most correct responses as most of the subjects correctly identified the injury, with an average accuracy of approximately 82%. However, in most cases concerning leg injuries, most participants who indicated that the upper leg was injured also perceived the lower leg as injured, regardless of whether both parts of the leg were actually injured.

TABLE III.
QUESTIONNAIRE : RESULTS FOR THE 1ST VIDEO – LEFT LOWER LEG INJURED

Q1	Did you notice a change in the animation? (yes/no question)	**Yes:10 (100%)**			No:0 (0%)	
Q2	How would you rate the fluidity of the transition?	Very Poor: 0 (0%)	Poor: 1 (10%)	Fair: 1 (10%)	Good: 0 (0%)	**Very Good: 8 (80%)**
Q3	How would you rate the progress in the transition?	Very Poor: 0 (0%)	Poor: 1 (10%)	Fair: 0 (0%)	Good: 4 (40%)	Very Good: 5 (50%)
Q4	Did you find the resulting animation realistic?	Very Poor: 0 (0%)	Poor: 2 (20%)	Fair: 1 (10%)	Good: 1 (10%)	**Very Good: 6 (60%)**
Q5	Which part of the body do you think is injured? (MCQ)	Head: 0 (0%)	Arm: 1 (10%)	Stomach: 0 (0%)	**Upper Leg: 8 (80%)**	**Lower Leg: 9 (90%)**

TABLE IV.
QUESTIONNAIRE : RESULTS FOR THE 2ND VIDEO – RIGHT UPPER LEG INJURED

Q1	Did you notice a change in the animation? (yes/no question)	**Yes:9 (90%)**			No:1 (10%)	
Q2	How would you rate the fluidity of the transition?	Very Poor: 0 (0%)	Poor: 0 (0%)	Fair: 1 (10%)	Good: 1 (10%)	**Very Good: 8 (80%)**
Q3	How would you rate the progress in the transition?	Very Poor: 0 (0%)	Poor: 0 (0%)	Fair: 1 (10%)	Good: 2 (20%)	**Very Good: 7 (70%)**
Q4	Did you find the resulting animation realistic?	Very Poor: 0 (0%)	Poor: 0 (0%)	Fair: 2 (20%)	Good: 2 (20%)	**Very Good: 6 (60%)**
Q5	Which part of the body do you think is injured? (MCQ)	Head: 0 (0%)	Arm: 1 (10%)	Stomach: 0 (0%)	**Upper Leg: 8 (80%)**	**Lower Leg: 8 (80%)**

TABLE V.
QUESTIONNAIRE : RESULTS FOR THE 3RD VIDEO – RIGHT LOWER LEG INJURED

Q1	Did you notice a change in the animation? (yes/no question)	**Yes:10 (100%)**			No:0 (0%)	
Q2	How would you rate the fluidity of the transition?	Very Poor: 1 (10%)	Poor: 1 (10%)	Fair: 2 (20%)	Good: 3 (30%)	Very Good: 3 (30%)
Q3	How would you rate the progress in the transition?	Very Poor: 1(10%)	Poor: 1 (10%)	Fair: 2 (20%)	Good: 3 (30%)	Very Good: 3 (30%)
Q4	Did you find the resulting animation realistic?	Very Poor: 1 (10%)	Poor: 2 (20%)	Fair: 1 (10%)	Good: 2 (20%)	Very Good: 4 (40%)
Q5	Which part of the body do you think is injured? (MCQ)	Head: 0 (0%)	Arm: 1 (10%)	Stomach: 1 (10%)	**Upper Leg: 7 (70%)**	**Lower Leg: 7 (70%)**

TABLE VI.
QUESTIONNAIRE : RESULTS FOR THE 4TH VIDEO – LEFT FOREARM INJURED

Q1	Did you notice a change in the animation? (yes/no question)	**Yes:10 (100%)**			No:0 (0%)	
Q2	How would you rate the fluidity of the transition?	Very Poor: 3 (30%)	Poor: 3 (30%)	Fair: 1 (10%)	Good: 1 (10%)	Very Good: 2 (20%)
Q3	How would you rate the progress in the transition?	Very Poor: 3 (30%)	Poor: 3 (30%)	Fair: 2 (20%)	Good: 0 (0%)	Very Good: 2 (20%)
Q4	Did you find the resulting animation realistic?	Very Poor: 4 (40%)	Poor: 2 (20%)	Fair: 1 (10%)	Good: 0 (0%)	Very Good: 3 (30%)
Q5	Which part of the body do you think is injured? (MCQ)	Head: 0 (0%)	**Arm: 10 (100%)**	Stomach: 1 (10%)	Upper Leg: 0 (%)	Lower Leg: 0 (%)

TABLE VII.
QUESTIONNAIRE : RESULTS FOR THE 5TH VIDEO – LEFT LOWER LEG /RIGHT UPPER LEG INJURED

Q1	Did you notice a change in the animation? (yes/no question)	**Yes:8 (80%)**			No:2 (20%)	
Q2	How would you rate the fluidity of the transition?	Very Poor: 1 (10%)	Poor: 1 (10%)	Fair: 2 (20%)	Good: 0 (0%)	**Very Good: 6 (60%)**
Q3	How would you rate the progress in the transition?	Very Poor: 1 (10%)	Poor: 2 (20%)	Fair: 3 (30%)	Good: 1 (10%)	Very Good: 3 (30%)
Q4	Did you find the resulting animation realistic?	Very Poor: 0 (0%)	Poor: 2 (20%)	Fair: 2 (20%)	Good: 2 (20%)	Very Good: 4 (40%)
Q5	Which part of the body do you think is injured? (MCQ)	Head: 0 (0%)	Arm: 1 (10%)	Stomach: 0 (0%)	**Upper Leg: 8 (80%)**	**Lower Leg: 8 (80%)**

TABLE VIII.
QUESTIONNAIRE : RESULTS FOR THE 6TH VIDEO – LEFT LOWER LEG /RIGHT UPPER LEG INJURED

Q1	Did you notice a change in the animation? (yes/no question)	**Yes:5 (62.5%)**			No:3 (37.5%)	
Q2	How would you rate the fluidity of the transition?	Very Poor: 1 (10%)	Poor: 1 (10%)	Fair: 2 (20%)	Good: 1 (10%)	Very Good: 5 (50%)
Q3	How would you rate the progress in the transition?	Very Poor: 1 (10%)	Poor: 1 (10%)	Fair: 2 (20%)	Good: 2 (20%)	Very Good: 4 (40%)
Q4	Did you find the resulting animation realistic?	Very Poor: 0 (0%)	Poor: 1 (10%)	Fair: 0 (0%)	Good: 5 (50%)	Very Good: 4 (40%)
Q5	Which part of the body do you think is injured? (MCQ)	Head: 0 (0%)	Arm: 1 (10%)	Stomach: 0 (0%)	**Upper Leg: 6 (60%)**	**Lower Leg: 9 (90%)**

TABLE IX.
QUESTIONNAIRE : RESULTS FOR THE 7TH VIDEO – LEFT LOWER LEG /RIGHT LOWER LEG INJURED

Q1	Did you notice a change in the animation? (yes/no question)	**Yes:7 (70%)**			No:3 (30%)	
Q2	How would you rate the fluidity of the transition?	Very Poor: 0 (0%)	Poor: 1 (10%)	Fair: 3 (30%)	Good: 4 (40%)	Very Good: 2 (20%)
Q3	How would you rate the progress in the transition?	Very Poor: 1 (10%)	Poor: 2 (20%)	Fair: 3 (30%)	Good: 3 (30%)	Very Good: 1 (10%)
Q4	Did you find the resulting animation realistic?	Very Poor: 0 (0%)	Poor: 3 (30%)	Fair: 0 (0%)	Good: 5 (50%)	Very Good: 2 (20%)
Q5	Which part of the body do you think is injured? (MCQ)	Head: 0 (0%)	Arm: 1 (10%)	Stomach: 0 (0%)	**Upper Leg: 8 (80%)**	**Lower Leg: 8 (80%)**

TABLE X. E
QUESTIONNAIRE : RESULTS FOR THE 8TH VIDEO – LEFT LOWER LEG /RIGHT LOWER LEG INJURED

Q1	Did you notice a change in the animation? (yes/no question)	**Yes:7 (70%)**			No:3 (30%)	
Q2	How would you rate the fluidity of the transition?	Very Poor: 0 (0%)	Poor: 2 (20%)	Fair: 3 (30%)	Good: 2 (20%)	Very Good: 3 (30%)
Q3	How would you rate the progress in the transition?	Very Poor: 1 (10%)	Poor: 2 (20%)	Fair: 1 (10%)	Good: 5 (50%)	Very Good: 1 (10%)
Q4	Did you find the resulting animation realistic?	Very Poor: 0 (0%)	Poor: 1 (10%)	Fair: 2 (20%)	Good: 3 (30%)	Very Good: 4 (40%)
Q5	Which part of the body do you think is injured? (MCQ)	Head: 0 (0%)	Arm: 1 (10%)	Stomach: 0 (0%)	**Upper Leg: 8 (80%)**	**Lower Leg: 9 (90%)**

TABLE XI. E
QUESTIONNAIRE : RESULTS FOR THE 9TH VIDEO – RIGHT UPPER LEG/RIGHT LOWER LEG INJURED

Q1	Did you notice a change in the animation? (yes/no question)	**Yes:7 (70%)**			No:3 (30%)	
Q2	How would you rate the fluidity of the transition?	Very Poor: 0 (0%)	Poor: 2 (20%)	Fair: 2 (20%)	Good: 1 (10%)	Very Good: 5 (50%)
Q3	How would you rate the progress in the transition?	Very Poor: 0 (0%)	Poor: 1 (10%)	Fair: 2 (20%)	Good: 3 (30%)	Very Good: 4 (40%)
Q4	Did you find the resulting animation realistic?	Very Poor: 0 (0%)	Poor: 1 (10%)	Fair: 2 (20%)	Good: 2 (20%)	Very Good: 5 (50%)
Q5	Which part of the body do you think is injured? (MCQ)	Head: 0 (0%)	Arm: 1 (10%)	Stomach: 0 (0%)	**Upper Leg: 8 (80%)**	**Lower Leg: 9 (90%)**

TABLE XII. E
QUESTIONNAIRE : RESULTS FOR THE 10TH VIDEO – LEFT LOWER LEG INJURED

Q1	Did you notice a change in the animation? (yes/no question)	**Yes:8 (80%)**			No:2 (20%)	
Q2	How would you rate the fluidity of the transition?	Very Poor: 1 (10%)	Poor: 2 (20%)	Fair: 0 (0%)	Good: 2 (20%)	Very Good: 5 (50%)
Q3	How would you rate the progress in the transition?	Very Poor: 1 (10%)	Poor: 3 (30%)	Fair: 1 (10%)	Good: 2 (20%)	Very Good: 3 (30%)
Q4	Did you find the resulting animation realistic?	Very Poor: 0 (0%)	Poor: 1 (10%)	Fair: 3 (30%)	Good: 2 (20%)	Very Good: 4 (40%)
Q5	Which part of the body do you think is injured? (MCQ)	Head: 0 (0%)	Arm: 1 (10%)	Stomach: 0 (0%)	**Upper Leg: 9 (90%)**	**Lower Leg: 9 (90%)**

TABLE XIII. E
QUESTIONNAIRE : RESULTS FOR THE 11TH VIDEO – RIGHT UPPER LEG INJURED

Q1	Did you notice a change in the animation? (yes/no question)	**Yes:8 (80%)**			No:2 (20%)	
Q2	How would you rate the fluidity of the transition?	Very Poor: 1 (10%)	Poor: 1 (10%)	Fair: 3 (30%)	Good: 3 (30%)	Very Good: 2 (20%)
Q3	How would you rate the progress in the transition?	Very Poor: 1 (10%)	Poor: 4 (40%)	Fair: 1 (10%)	Good: 4 (40%)	Very Good: 0 (0%)
Q4	Did you find the resulting animation realistic?	Very Poor: 2 (20%)	Poor: 2 (20%)	Fair: 1 (10%)	Good: 1 (10%)	Very Good: 4 (40%)
Q5	Which part of the body do you think is injured? (MCQ)	Head: 0 (0%)	Arm: 2 (22%)	Stomach: 1 (10%)	**Upper Leg: 7 (70%)**	**Lower Leg: 5 (50%)**

TABLE XIV.
QUESTIONNAIRE : RESULTS FOR THE 12TH VIDEO – RIGHT LOWER LEG INJURED

Q1	Did you notice a change in the animation? (yes/no question)	**Yes:8 (88.9%)**			No:1 (11.1%)	
Q2	How would you rate the fluidity of the transition?	Very Poor: 1 (10%)	Poor: 3 (30%)	Fair: 0 (0%)	Good: 3 (30%)	Very Good: 3 (30%)
Q3	How would you rate the progress in the transition?	Very Poor: 2 (20%)	Poor: 4 (40%)	Fair: 1 (10%)	Good: 3 (30%)	Very Good: 0 (0%)
Q4	Did you find the resulting animation realistic?	Very Poor: 2 (20%)	Poor: 3 (30%)	Fair: 2 (20%)	Good: 1 (10%)	Very Good: 2 (20%)
Q5	Which part of the body do you think is injured? (MCQ)	Head: 0 (0%)	Arm: 0 (0%)	Stomach: 0 (0%)	**Upper Leg: 10 (100%)**	**Lower Leg: 7 (70%)**

TABLE XV.
QUESTIONNAIRE : RESULTS FOR THE 13TH VIDEO – LEFT FOREARM INJURED

Q1	Did you notice a change in the animation? (yes/no question)	Yes:3 (30%)			**No:7 (70%)**	
Q2	How would you rate the fluidity of the transition?	**Very Poor: 4 (40%)**	Poor: 2 (20%)	Fair: 2 (20%)	Good: 1 (10%)	Very Good: 1 (10%)
Q3	How would you rate the progress in the transition?	**Very Poor: 5 (50%)**	**Poor: 4 (40%)**	Fair: 1 (10%)	Good: 0 (0%)	Very Good: 0 (0%)
Q4	Did you find the resulting animation realistic?	**Very Poor: 4 (40%)**	**Poor: 5 (50%)**	Fair: 0 (0%)	Good: 0 (0%)	Very Good: 1 (10%)
Q5	Which part of the body do you think is injured? (MCQ)	Head: 0 (0%)	**Arm: 10 (100%)**	Stomach: 0 (0%)	Upper Leg: 0 (0%)	Lower Leg: 0 (0%)

TABLE XVI.
ANIMATION QUALITY QUESTIONNAIRE RESULTS

	Average	Standard deviation
Q1	78.57%	19%
Q2	3.6	0.69
Q3	3.2	0.84
Q4	3.5	0.70

V. Discussion

Our first set of experiments demonstrated that the proposed system achieved most of our objectives. Based on the questionnaire results, we can conclude that our animation transitions are fluid and progressive, both in terms of the character speed and animation. Although the results are not optimum, the quality is consistent, and the process time is relatively low, critical considerations for videogame applications of this system.

TABLE XVII.
UNTIME SPEED DEPENDING ON THE PHASE FUNCTION COMPUTATION TECHNIQUE USED

Technique	Runtime (light experiment)	Runtime (normal experiment)
PFNN cubic	5.513 ms	5.258 ms
PFNN linear	1.364 ms	1.264 ms
PFNN constant	0.659 ms	0.755 ms

From the questionnaire experiment, we also learn that our system produces excellent results when generating animations in which a single body part is injured, and performs adequately well with multiple injured body parts. However, videos 4 and 13 did receive negative feedback. These videos depicted animations involving an injured arm. As mentioned previously, this may have been due to animation errors concerning one elbow joint resulting from the low quality training animation. More tests using other injured upper body part animations are required to verify this hypothesis.

It also seems that it was still difficult to distinguish the difference between the two lower body injuries to the same limb, the upper leg and lower leg. When the upper leg was injured, the character dragged the injured leg behind the rest of their body. When the lower leg was injured, the character felt pain as its foot contacted the ground. Although these two animations were distinct, most subjects were confused about which one body part was injured. In videos 5 through 9, both part of the leg were injured, but this was not the case with other videos. We suggest that gathering more distinguishable animation data could resolve this issue.

It also seems to be harder notifying the change in the animation when the character suffers multiple injuries. Having different animation for key injuries level (3 and 7 for example), It appeared to be more difficult for subjects to identify changes in the animation when the character suffers multiple injuries. Having different animations for key injury levels, such as levels 3 and 7, could make the animation evolution both more natural and more noticeable.

One obstacle that we encountered was that the running animation disrupted the original gait system. This must be resolved, either by revisiting our input format or by removing the running animation entirely and allowing the system to blend the injured walking animation and the uninjured running animation independently.

VI. Conclusion

We developed an approach that adds an injury information feature to a PFNN animation generation system. This feature changes character animation depending on body status and environment. Our method generates high quality animations when the lower body of the character body is injured. However, we must still address complications rendering upper body injuries.

We plan to further test our system after collecting additional animation data, particularly concentrating on the upper body and other animations with in-between severity values (3, 5 or 7) for each of the body parts. We also want to improve the quality of training set animations to correspondingly improve the accuracy of our results. We also need to test different scale coefficients of injury inputs in the NN to optimize the compromise between the smoothness of the animation and the range of injury expressed. With these additional efforts, we believe that this system will be suitable for character control in complex virtual situations such as video games.

Using our system at its full potential would require a large amount of animation data, because we need data for every potential combination of link animations, injury severities, motion gaits, and their extensions for every possible combination of different injuries. This contradicts one of our initial objectives of reducing reliance on animation data. To address this conflict, in future work we will consider isolating the animation of different body parts, to recreate a specific body status situation by combining the animations of different body parts. Finally, we also want to add new features to our system, starting with an "emotion" feature, that would alter the character behavior corresponding to emotional state.

ACKNOWLEDGMENT

We would like to thank Daniel Holden et al., creators of the original PFNN system and who graciously shared the specific 3D model we needed to retarget our animation data.

REFERENCES

[1] F.Petros, M.V, Panne and D.Terzopoulos, The virtual stuntman: Dynamic characters with a repertoire of autonomous motor skills. Computers & Graphics. 25. 933-953. 10.1016/S0097-8493(01)00171-6. 2009

[2] T. Geijtenbeek, N. Pronost , A. Egges and M. Overmars, Interactive character animation using simulated physics.2011

[3] Y. Abe, M.D. Silva and J.Popovic, Multi objective control with frictional contacts. 249-258. 10.1145/1272690.1272724.2007

[4] P. Beaudoin, S. Coros and M. V. Panne, Generalized biped walking control. ACM Trans. Graph., 29, 130:1-130:9.2011

[5] U. Muico, Y. Lee, J. Popović and Z. Popović, Contact-aware Nonlinear Control of Dynamic Characters. ACM Transactions on Graphics, 28, .2009

[6] T. Geijtenbeek, M. V. Panne and A. F. V. Stappen, Flexible Muscle-Based Locomotion for Bipedal Creatures. ACM Transactions on Graphics, 32, . 2013

[7] T. Geijtenbeek, D. Vasilescu and A. Egges, Injury Assessment for Physics-Based Characters. In: Allbeck J.M., Faloutsos P. (eds) Motion in Games. MIG 2011. Lecture Notes in Computer Science, Vol 7060. Springer, Berlin, Heidelberg.2011

[8] D. Charles and S. Mcglinchey, The past, present and future of artificial neural networks in digital games.2018

[9] J. Kim, F. Cordier and N. Magnenat-Thalmann, "Neural network-based violinist's hand animation," Proceedings Computer Graphics International 2000, Geneva, 2000, pp. 37-41.2000

[10] X. B. Peng, et al. "DeepLoco: dynamic locomotion skills using hierarchical deep reinforcement learning." ACM Trans. Graph. 36 (2017): 41:1-41:13.2017

[11] N. Heess, T.B. Dhruva, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, A. Eslami, M. Riedmiller and D. Silver, Emergence of Locomotion Behaviours in Rich Environments.2017

[12] D. Holden, J. Saito and T. Komura, A deep learning framework for character motion synthesis and editing. ACM Transactions on Graphics. 35. 1-11. 10.1145/2897824.2925975.2017

[13] D. Holden, T. Komura and J. Saito, Phase-functioned neural networks for character control. ACM Trans. Graph., 36, 42:1-42:13.2017

[14] C. L. von Baeyer, L. J. Spagrud, J.C. McCormick, E. Choo, K. Neville and M. A. Connelly, Three new datasets supporting use of the Numerical Rating Scale (NRS-11) for children's self-reports of pain intensity. Pain.; 143(3): 223-7.2009

[15] P. E. Bijur, C. T. Latimer, and E. J. Gallagher, Validation of a Verbally Administered Numerical Rating Scale of Acute Pain for Use in the Emergency Department. Academic Emergency Medicine, 10: 390–392. doi:10.1111/j.1553-2712.2003.tb01355.x .2003