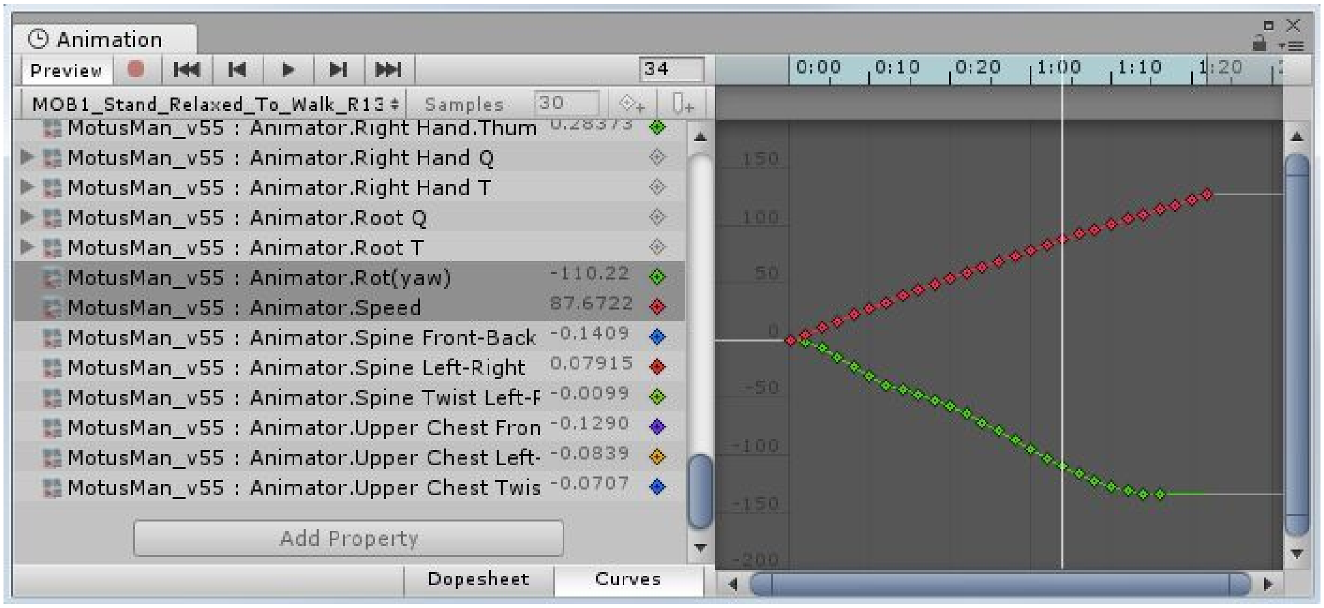
Unity IPC Custom Animation Curves - MocapOnline

Some of our Animation Packs now have **In-Place Custom(IPC) curves**. They can be ignored, and will not affect any animation programming you normally do and are accustomed to working with.
HOWEVER, combined with the **“Reference Root”** motion we offer(please see our other docs about Root motion) they are a potentially powerful and useful tool.

WHAT ARE IPC CURVES?



MOB1_Stand_Relaxed_To_Walk_R13

Samples 30

MotusMan_v55 : Animator.Right Hand.Thum	0.28373	◆
MotusMan_v55 : Animator.Right Hand Q		◇
MotusMan_v55 : Animator.Right Hand T		◇
MotusMan_v55 : Animator.Root Q		◇
MotusMan_v55 : Animator.Root T		◇
MotusMan_v55 : Animator.Rot(yaw)	-110.22	◆
MotusMan_v55 : Animator.Speed	87.6722	◆
MotusMan_v55 : Animator.Spine Front-Back	-0.1409	◆
MotusMan_v55 : Animator.Spine Left-Right	0.07915	◆
MotusMan_v55 : Animator.Spine Twist Left-F	-0.0099	◆
MotusMan_v55 : Animator.Upper Chest Fron	-0.1290	◆
MotusMan_v55 : Animator.Upper Chest Left-	-0.0839	◆
MotusMan_v55 : Animator.Upper Chest Twis	-0.0707	◆

Fig. 1

IPC curves are derived and calculated from the original **“Root”** animation as two values:

- **“Speed”** - The character’s cm/sec(centimeters per second) speed, frame by frame.
- **“Rot(yaw)”** - The character’s rotational heading(facing direction) in degrees, frame by frame.
  An animation always starts at zero degrees.

A simple example: **“Walk_Forward_Loop_IPC”** would have a constant **Speed** of 128.194 cm/sec, and a constant **Rot(yaw)** of zero degrees moving in a straight line.

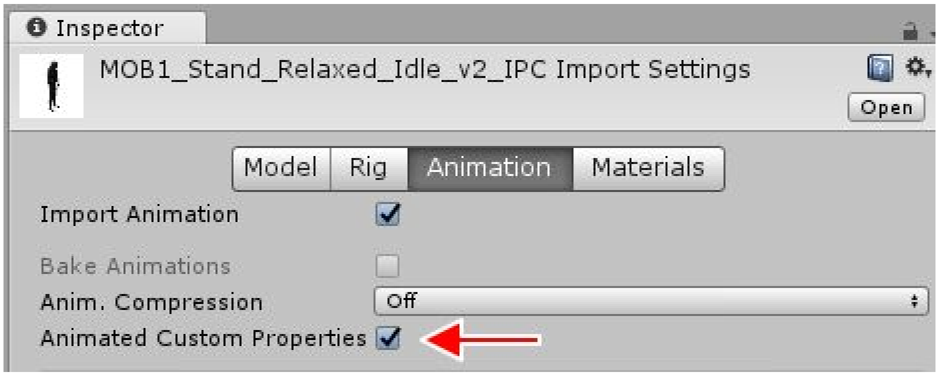
A more **complex example**, however:
Using (Fig. 1) above **“Stand_Relaxed_To_Walk_R135_Fwd_IPC”** as an example, the action result would be:

- The **“Root”** moves in a straight diagonal line Right/Back(at -135 deg) using the red curve values frame by frame for speed.
- The **“Root”** rotates to the right using the green curve values frame by frame for the number of degrees.
- The animation ends at frame 52, with the **Speed** at 126.388 cm/sec, and the **Rot(yaw)** heading at -135 degrees.
- This transition animation ends with the character walking in a straight line facing 135 degrees to the right, and would blend into **“Walk_Forward_Loop_IPC”** with a reset Rot(yaw) value of zero degrees, and a constant Speed value of 128.194 cm/sec.

By using these **“IPC”** curve values with **“Root”** motion, an in-place motion can be moved and accurately replicate the original variable root motion of the character. Instead of assigning static, linear or arbitrary values to the speed and/or heading as guesswork and creating foot sliding errors and other poor looking behavior.

IPC GETTING STARTED - Basic Settings

Make sure **“Animated Custom Properties”** is checked in the **Animation Import Settings** for all animations. (See Fig. 2)



Import Animation ☑

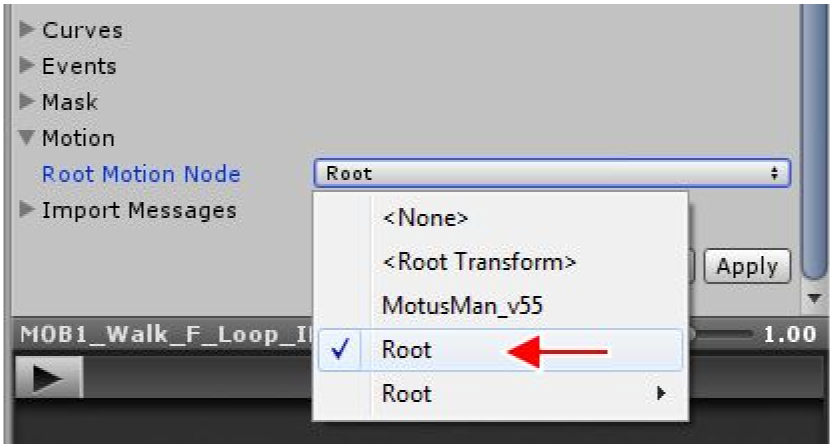
Bake Animations □

Anim. Compression Off

Animated Custom Properties ☑

Fig. 2

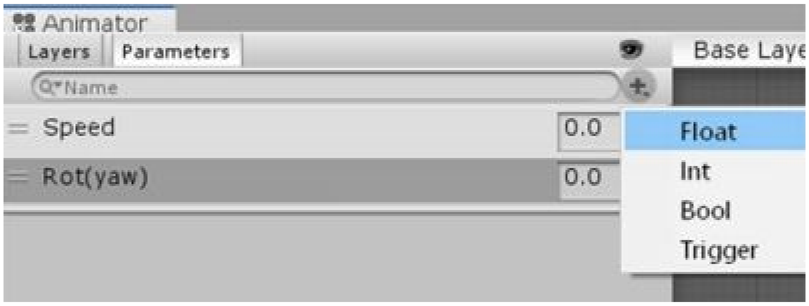
Since the IPC curves are based on the reference “Root” animation, in order to work correctly you have to set the character’s root motion node to **"Root"**.
You will find this setting right at the bottom of the **Animation Import Settings**. (See Fig. 3)



Root Motion Node Root

Fig. 3

To make them appear in the **“Animation” Window**, you have to add the respective parameters as **floats** to the **Animator** component. (See Fig. 4).



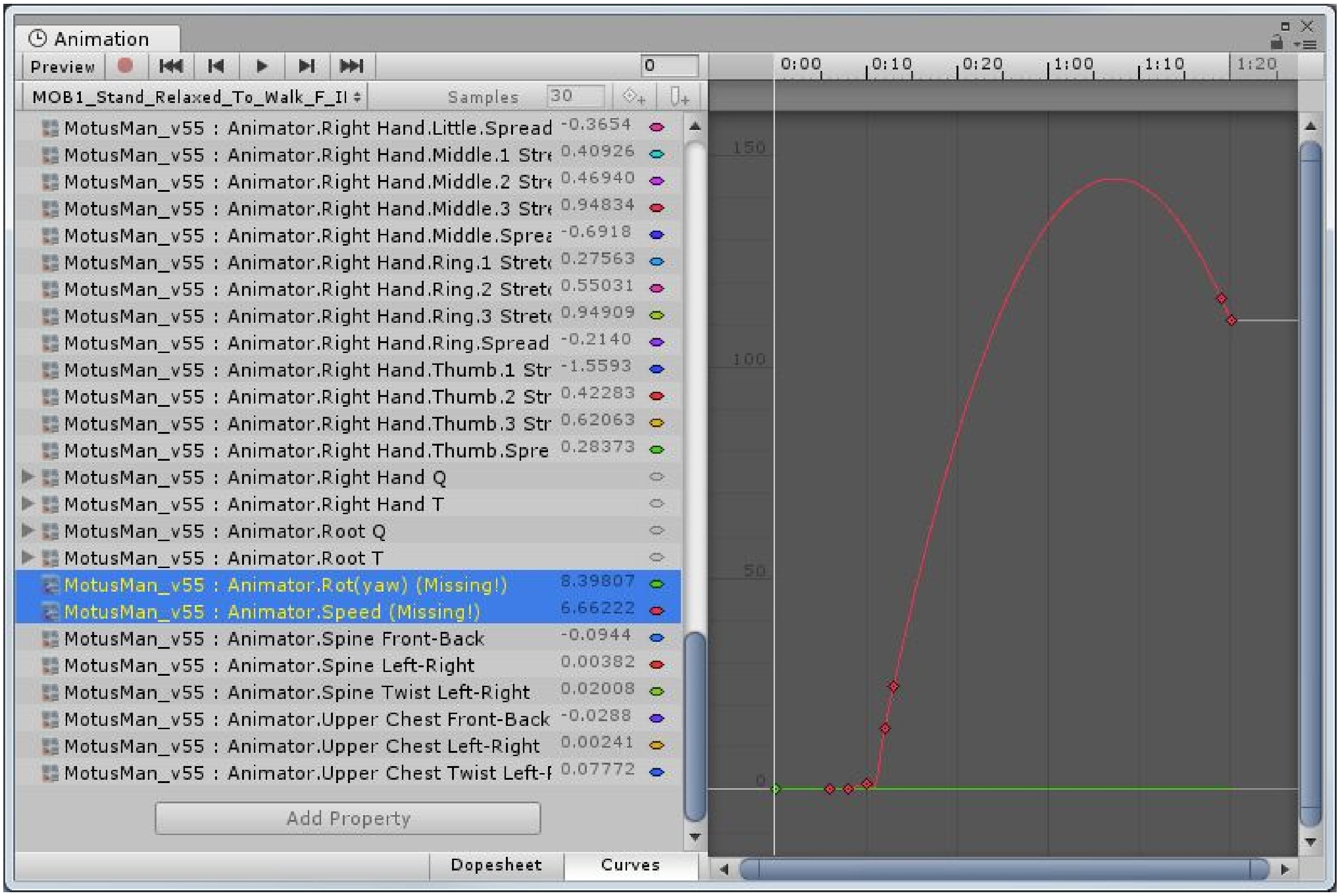
Speed 0.0

Rot(yaw) 0.0

Fig. 4

The naming has to exactly match the curve names, otherwise, it won‘t work. If you mistyped the parameter names or forgot to add them at all, they will appear to be **“Missing!”** in the “Animation” window (See Fig. 5). Make sure you added the parameters and their names match exactly.

NOTE: If Unity still shows “Missing!” in the animation window, you need to update the window by dragging the time slider on the right just a few frames. The “Missing!” notify should be gone now.



MotusMan_v55 : Animator.Right Hand.Little.Spread	-0.3654
MotusMan_v55 : Animator.Right Hand.Middle.1 Stre	0.40926
MotusMan_v55 : Animator.Right Hand.Middle.2 Stre	0.46940
MotusMan_v55 : Animator.Right Hand.Middle.3 Stre	0.94834
MotusMan_v55 : Animator.Right Hand.Middle.Sprea	-0.6918
MotusMan_v55 : Animator.Right Hand.Ring.1 Stret	0.27563
MotusMan_v55 : Animator.Right Hand.Ring.2 Stret	0.55031
MotusMan_v55 : Animator.Right Hand.Ring.3 Stret	0.94909
MotusMan_v55 : Animator.Right Hand.Ring.Spread	-0.2140
MotusMan_v55 : Animator.Right Hand.Thumb.1 Str	-1.5593
MotusMan_v55 : Animator.Right Hand.Thumb.2 Str	0.42283
MotusMan_v55 : Animator.Right Hand.Thumb.3 Str	0.62063
MotusMan_v55 : Animator.Right Hand.Thumb.Spre	0.28373
▶ MotusMan_v55 : Animator.Right Hand Q	
▶ MotusMan_v55 : Animator.Right Hand T	
▶ MotusMan_v55 : Animator.Root Q	
▶ MotusMan_v55 : Animator.Root T	
MotusMan_v55 : Animator.Rot(yaw) (Missing!)	8.39807
MotusMan_v55 : Animator.Speed (Missing!)	6.66222
MotusMan_v55 : Animator.Spine Front-Back	-0.0944
MotusMan_v55 : Animator.Spine Left-Right	0.00382
MotusMan_v55 : Animator.Spine Twist Left-Right	0.02008
MotusMan_v55 : Animator.Upper Chest Front-Back	-0.0288
MotusMan_v55 : Animator.Upper Chest Left-Right	0.00241
MotusMan_v55 : Animator.Upper Chest Twist Left-F	0.07772

Fig. 5

To read the parameters using **C# Script**, just use **GetFloat(“Speed”)** as mentioned in the **Unity docs:** https://docs.unity3d.com/ScriptReference/Animator.GetFloat.html

It is recommended to use 1/100 of the speed, as IPC curves are in centimeters per second, and Unity uses meters per second by default. You would need to use your own values, depending on your project settings.

```
voidStart()
{
  anim=gameObject.GetComponent<Animator>();
}

voidUpdate()
{
  Debug.Log(anim.GetFloat("Speed")/100);
}
```

***NOTE: This is not meant to be a full programming tutorial for using IPC curves!

It is just some basics on how to access and initially approach using **IPC**, thanks to our clever friend **Zyblade** on the official **Unity Community Forum**!

Zyblade saw the **IPC** curves we include in **UE4** and asked us to offer them in **Unity** as well. He has done some very cool things with them!
Please direct any detailed programming conversation to the **Unity Community Forum**.
Zyblade and **MocapOnline** will be there as well to discuss the usage of **IPC** curves.

www.MocapOnline.com
https://assetstore.unity.com/publishers/4746
Unity Community Forum
Unity@MotusDigital.com