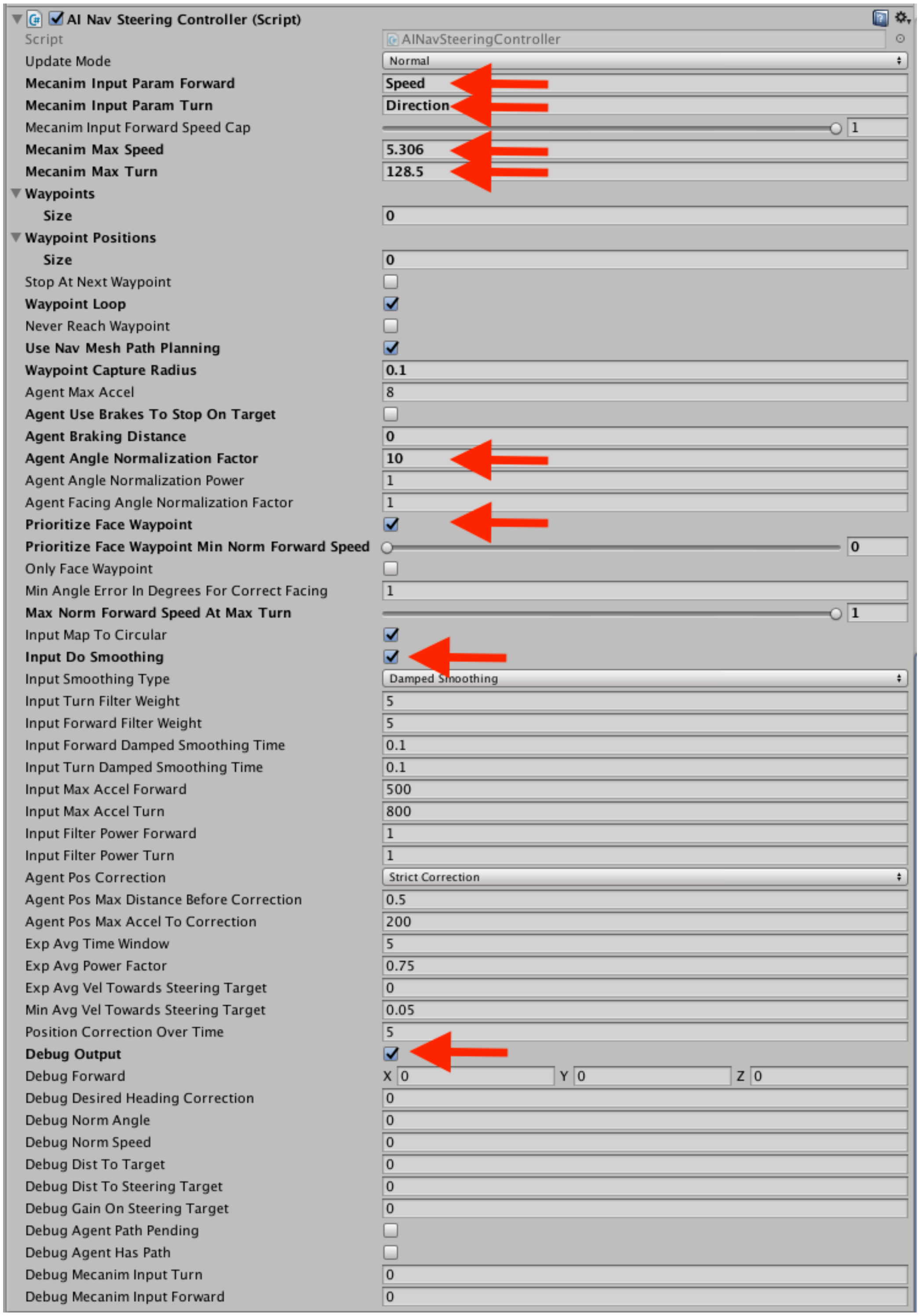
AINavSteering Controller: How to Tune for your Mecanim Character

I highly recommend checking out the demo project at
https://github.gatech.edu/IMTC/CS4455_MecanimTute

In that project is a CS4455_AI_Demo scene with an AINavSteeringController script that is designed to work with generic Mecanim Animators that support variable forward translation and turning. Specifically, the controller supports Mecanim characters with a forward Animator parameter at least in the range [0, 1] (range [-1,1] is fine too) with 0 as stopped and 1 as full speed ahead. Additionally, the Mecanim character needs a turn Animator parameter in the range [-1, 1] with -1 full left turn, 0 no turn, and 1 full right turn. While not required, it's highly recommended that your character support turning in place and smoothly blending from turning in place to turning while going forward. If your character meets these criteria, then it should theoretically work fine with AINavSteeringController.

There are a few steps required to get AINavSteeringController up and running.
1.) Configure a NavMesh properly in your scene (make sure you see blue in navigable areas in the Navigator view).
2.) Add your character to a navigable spot in the scene
3.) Disable or remove any user input-based controller script from your Mecanim character
4.) I recommend that you set your Animator to Update Mode: Animate Physics. However, this isn't necessary.
5.) Add AINavSteeringController to your character
6.) In the script for AINavSteeringController, if you have a CharacterController instead of RigidBody+Capsule then uncomment #define USE_CHARACTER_CONTROLLER

Here is a screen cap of the AINavSteeringController component Inspector view. Refer to it as you read the remaining steps.



☑ AI Nav Steering Controller (Script)	
Script	AINavSteeringController
Update Mode	Normal
Mecanim Input Param Forward	Speed
Mecanim Input Param Turn	Direction
Mecanim Input Forward Speed Cap	1
Mecanim Max Speed	5.306
Mecanim Max Turn	128.5
Waypoints	
Size	0
Waypoint Positions	
Size	0
Stop At Next Waypoint	☐
Waypoint Loop	☑
Never Reach Waypoint	☐
Use Nav Mesh Path Planning	☑
Waypoint Capture Radius	0.1
Agent Max Accel	8
Agent Use Brakes To Stop On Target	☐
Agent Braking Distance	0
Agent Angle Normalization Factor	10
Agent Angle Normalization Power	1
Agent Facing Angle Normalization Factor	1
Prioritize Face Waypoint	☑
Prioritize Face Waypoint Min Norm Forward Speed	0
Only Face Waypoint	☐
Min Angle Error In Degrees For Correct Facing	1
Max Norm Forward Speed At Max Turn	1
Input Map To Circular	☑
Input Do Smoothing	☑
Input Smoothing Type	Damped Smoothing
Input Turn Filter Weight	5
Input Forward Filter Weight	5
Input Forward Damped Smoothing Time	0.1
Input Turn Damped Smoothing Time	0.1
Input Max Accel Forward	500
Input Max Accel Turn	800
Input Filter Power Forward	1
Input Filter Power Turn	1
Agent Pos Correction	Strict Correction
Agent Pos Max Distance Before Correction	0.5
Agent Pos Max Accel To Correction	200
Exp Avg Time Window	5
Exp Avg Power Factor	0.75
Exp Avg Vel Towards Steering Target	0
Min Avg Vel Towards Steering Target	0.05
Position Correction Over Time	5
Debug Output	☑
Debug Forward	X 0 Y 0 Z 0
Debug Desired Heading Correction	0
Debug Norm Angle	0
Debug Norm Speed	0
Debug Dist To Target	0
Debug Dist To Steering Target	0
Debug Gain On Steering Target	0
Debug Agent Path Pending	☐
Debug Agent Has Path	☐
Debug Mecanim Input Turn	0
Debug Mecanim Input Forward	0

7.) Configure mecanimInputParamForward and mecanimInputParamTurn with the case-sensitive names of your Animator's parameters for forward and turning motion in Inspector view of AINavSteeringController.

8.) Inspect your fastest forward animation and your fastest forward turning animation for the

Average Velocity (m/sec) and Average Angular Y Speed (deg/sec). If you select these animations in your Project view, the Inspector view will show the averages near the bottom (possibly in partially grayed out text). Manually set these values to mecanimMaxSpeed and mecanimMaxTurn of AINavSteeringController in the Inspector.
9.) Now create several navigable waypoint geometries in your scene and remove the colliders, but leave the visuals (e.g. a box)
10.) In the Inspector view of AINavSteeringController, increase the Size of the Waypoints array and assign each slot one of your waypoint geometries.
11.) You now have enough in place to begin tuning the AINavSteeringController in real-time. However, to make it easier to see what you are doing you should uncheck inputDoSmoothing and check debugOutput.
12.) Now hit play. You should see your character move but probably VERY jerky. (During the next tuning steps, sometimes your character will go off a cliff, become stuck, or otherwise go past the point of no return. In these cases, just stop and restart.)
13.) At this point, you primarily need to focus on tuning the turning such that it isn't going full left to full right. You want to adjust agentAngleNormalizationFactor larger and larger until you can get variable turns that reasonably head in the direction of the current waypoint. I recommend you look at your Animator view as the character moves about. Keep increasing agentAngleNormalizationFactor until you no longer see -1 or 1 for the turn param (note that these values will instead be unit circle positions if inputMapToCircular is set to true). Now decrease just until you are back to seeing some -1s and 1s for the hardest turns (and values in between for slight turns). Additionally, look at the Scene view at the blue and green ray (turned on by debugOutput setting). You want to see the blue ray quickly turning towards the green, and ultimately aligning with it when your character is walking straight. Ideally you want to rotations of the blue ray avoid overshooting and going back and forth.
14.) Assuming you did well with the step above, make note of your final agentAngleNormalizationFactor and type it in after you stop the game and lose your setting.
15.) Now turn on inputDoSmoothing and play the scene again. The turning should be pretty smooth. If not, check out the tooltip documentation on the input smoothing params and make adjustments.
16.) Assuming you have lots of obstacles in your scene, the next thing to look for is whether your character's turning radius is good enough to avoid running into everything. You may find that your character needs to slow down to make turns properly. They way to do this is to set prioritizeFaceWaypoint to true. This will force your character to first turn in place towards the waypoint until it is at least +/- 180 degrees difference from the forward direction. Then, as this angle gets smaller the character will go progressively faster in the forward direction.
17.) The above steps detail the most important settings for dialing in the character, however there is lots more that can be tweaked. Just check out the tooltips in the Inspector.
18.) Lastly, check out AIDemoController for an example state machine that talks to the AINavSteeringController