Andrew Rozniakowski

Math441

4/28/16

Movement Imitation with Nonlinear Dynamical Systems in Humanoid Robots

For this project I studied a paper written by Jan Ijspeert, Jun Nakanishi and Stefan Schaal. The authors of this paper created a nonlinear dynamical system that imitates human movements in autonomous humanoid robots in-order to help and study stroke victims. This means their dynamical system doesn’t just have the robot copy the movements of the human but they learn the movement as well as others. This allows for the robot to make real time adjustments in order to avoid perturbations or changes to the environment. Their system works as a control policy which sends signals to the controller of the robot to tell it to move. The authors choose to represent movements in kinematic coordinates such as joint angles of the robot. Therefore, the control policy represents kinematic movement plans where a change of position is thought of as a change from one state to another. They then assumed the robots controller can turn these movement plans into motor commands, thus making the robot move. To accomplish their goals the authors had to use two separate dynamical systems. For this project I only studied the stability and phase portraits of their main system, however, first I will give a brief summary on both systems.

The main system of the paper, the one I chose to study, is the control policy for the robot. It sends signals to the robots controller in-order for the robot to move. The system is: (1)

. (2)

The constants in this equation, and, are always greater than 0. The variable g is the goal position for the robot. The position or state the robot wants to get to. The variable y is the desired position that gets outputted to the controller of the robot while is the desired velocity. The system is essentially a second order system however, the variable v, which is velocity, is modified by a nonlinear term. Subsequently, the nonlinear term depends on the internal states of the robot since the sign and value of v can change based on where the robot is facing and which direction it’s moving in. Thus, the second order linear system given by:

(3)

, (4)

was created to describe the internal states of the robot. Once again the constants, and are greater than 0. The variable v is a scaling term used to ensure the nonlinear term in equation 2 remains transient. Meaning, v must start and end at 0 during the discrete movement. The variable x is used to localize the nonlinear term in equation 2 of the first system. The nonlinear term,, is created using the Gaussian kernel:

. (5)

The term also needs a variable which can be found using locally weighted regression. This is an important part of the system because it is the part that allows the robot to not just mimic movements but also learn them. Gaussian kernels are often used in machine learning projects. Now that we understand the systems we can study its stability.

In order to study the stability of the system made up from equations 1 and 2 we must look at the system as time approaches infinity. Since v is transient, when we study the equations as they go to infinity, v will approach 0. Therefore, the nonlinear term will also go to 0, leaving us with the system:

(6)

(7)

This system has exactly one fixed point, which is (0, g). To study the stability of this fixed point we must linearize the system by finding its Jacobian matrix. Therefore, we must find the partial derivatives of equations 6 and 7 with respect to both y and z. We start with equation 6 which can be re-written as:

. (8)

We then find our partial derivative with respect to z and y.

,

.

Next we do the same for .

,

.

Thus our Jacobian matrix is:

.

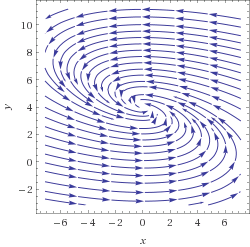
We then evaluate the Jacobian at our fixed point (0, g) but it does not affect our matrix. Next we must solve for our matrix’s eigenvalues. First we find the determinant which is:

We then solve for our eigenvalues which are:

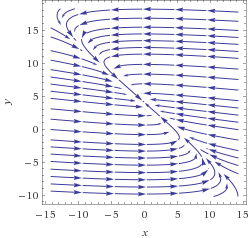
, (9)

. (10)

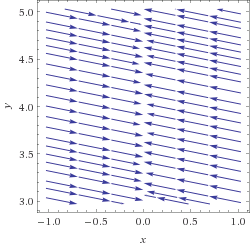
This gives us three different cases to consider when studying the stability. The first is when . In this case we will end up with imaginary eigenvalues. Since the α in our complex eigenvalue does not equal 0 we will have a spiraling phase portrait. Since our α is also < 0, since is always greater than 0, the phase portrait will be spiraling inwards into the fixed point (0, g). Thus, in this case we have a stable fixed point. The phase portrait can be seen below with the values and g = 4.



Next we must look at the case when . In this situation we will have real eigenvalues. Looking at equation 9 we see will always be negative. Therefore, the next two cases we must consider deals with the value of. When will be greater than 0. Thus, which gives us a saddle point phase portrait. Since the trajectories of our phase portrait will be moving towards the fixed point, once again giving us a stable fixed point. The phase portrait can be seen below with the values and g = 4.



The final case we must study is when . It is not possible for to be greater than however, it is possible for them to get infinitely close together. If these two terms are equal then = 0. Thus, our phase portrait will have an infinite line of fixed points vertically down the center at z = 0. Since < 0 the trajectories will once again be moving towards the fixed points. Thus, we have a stable fixed point. The phase portrait can be seen below using the values and g = 4.



In all three of our cases we not only have stable fixed points but it is also globally attracting since all trajectories end at the fixed point. These fixed points are also Liapunov stable since all the trajectories close to the fixed point will stay close to the fixed point. Thus, we can call our fixed points asymptotically stable.

On paper, the nonlinear dynamic system created by the authors of this paper accomplished the goals they set out to. Since their system will always have an asymptotically stable fixed point, no matter the parameters, this means no matter the initial state of the robot it will always be able to move to the goal position g. There can be issues with error when the robot actually moves, but the system will always send the correct movement plans to the robots controller. When testing their system the authors only tested movements involving the arm. They tested it using three different arm movements which are reaching for objects, drawing 2D pictures and imitating a tennis swing. In all three tests their dynamic system allowed for fast learning of trajectories as well as the ability to recover from perturbations. The authors hope to use their system to help stroke patients during rehabilitation exercises by showing them movements and modeling their movements.