INTELLIGENT CONTROL OF ROBOTIC SYSTEMS

Prof. M. Felix Orlando

Department of Electrical Engineering

Indian Institute of Technology Roorkee

Lecture 04: Feedback Linearization

Good morning everyone. Today we are going to learn about feedback linearization of robotic manipulators. The organization of this lecture will be as follows. First, we will see the feedback linearization approach and then apply it to robot manipulators for both kinematics-based control and dynamics-based control. Let us start.

Feedback linearization. So, what is that about? It is about making the closed-loop system dynamics of a nonlinear system becomes stable and linear through a suitable control law. If the closed-loop system dynamics become stable and linear through a suitable control law, then this approach is termed as feedback linearization. If the closed-loop system dynamics of a robotic system, for example, $\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + \underline{G(\theta)}$

is a system equation, which means it is a highly coupled nonlinear system dynamic equation for an n-degree of freedom robotic system. If this system representing the robotic system equation, representing the robotic system, by a suitable control law becomes stable and linear through a suitable control law, then we can say that we have applied the feedback linearization approach to the system. In fact, there exists a class of nonlinear systems.

There exists a class of non-linear systems for which feedback linearization can be applied. The approach can be applied. Let us consider a class of non-linear systems. Consider a class of single-input affine non-linear systems given by

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_n \\ f(x) + g(x)u \end{bmatrix}$$

Let this be represented by equation number 1.

So, we can say that the output equation is given by $y = x_1 \rightarrow (2)$

Assumption: $g(x) \neq 0$.

It is equation 2. So, which implies that g (x) is invertible.

So, if we choose the control input given by

$$u = \frac{-1}{g(x)}\left[-f(x) + k_v r + \lambda_1 e^{(n-1)} + \cdots + \lambda_{n-1} e^{(1)} + \cdots + x_{nd}\right]$$

where I mentioned where, error, $\boxed{e = y_d - y}$ is the o/p tracking error. e is nothing but output tracking error.

Y is the actual output of the system and Y desired is the desired value for the system and y d minus y is the tracking error and let us consider $r = e^{(n-1)} + \lambda_1 e^{(n-2)} + \cdots + \lambda_{n-1} e$

So, I indicate that $\dot{e} \Rightarrow e^{(1)}$

So, the power indicates a derivative. So, the closed-loop system dynamics precisely describe the error dynamics. The closed-loop error dynamical equation for this case becomes $\dot{r} = -k_v r$ which is linear as well as stable by choosing kv as positive. So, kv and lambdas are positive design parameters for the feedback linearization control approach. Thus, such a technique is called the feedback linearization approach, which stabilizes the closed-loop system dynamics. Now, let us consider an example on the first page.

Example 1. Let us consider a single-link robot manipulator whose dynamical equation is given by $\tau = ml^2\ddot{q} + b\dot{q} + mgl\sin q \Rightarrow (1)$

where q is the generalized coordinate of the robot. Here, we have a single-link robot manipulator, where the joint angle or the generalized coordinate is denoted by theta. It is a revolute-jointed robotic system. The system parameters are m equal to mass, which is equal to 1 kilogram.

The length of the robot is 1 meter. And we have the acceleration due to gravity given by g equal to 10 meters per second squared. With the b value, which is the term responsible for Coriolis and centrifugal forces, that is taken to be 0.1. So, for this system.

The question is to design a control law that would make the closed-loop error dynamics linear and stable. Let us see that now. So, the solution to this case is that it is desired for the manipulator to track a desired joint angle.

So, the objective is to track a desired joint. Joint angle given by q_d, and let us describe or define the error, called the tracking error as, $e = q_d - q$ where q_d is the desired angle minus the actual joint angle of the robotic system.

This will be equation 2. Let the control. Law be chosen as $\tau = a(\ddot{q}_d + k_D \dot{e} + k_p e) + b\dot{q} + c \sin q \rightarrow (3)$

Let this control law be represented by equation number 3. Next, we can see in the control law that which is the mass matrix, you can say. $a = ml^2$, $c = mgl$.

So, now with this control law of τ, let us replace the τ expression in the actual matrix. System dynamics that is given by equation 1. So, we get $a\ddot{q} + b\dot{q} + mgl \sin q = a(\ddot{q}_d + k_D \dot{e} + k_p e) + b\dot{q} + c \sin q$ So, this is the situation.

Now, we can say that this can be equation number 4. So, thus, equation 4 is going to be reduced. Okay. So, you can see that upon simplification, equation 4 turns out to be a second-order error dynamic equation given by $\ddot{e} + k_D \dot{e} + k_p e = 0$ which is equation number 5.

Thus, The closed-loop error dynamics is linear and stable as well for positive values of the controller gains K_p and K_d, okay. So, that is the situation here, okay. So, we have finished the first example in this case using a dynamic model of a robotic system. We can now move into the feedback linearization control approach towards the kinematic model of a robotic system.

We now have a kinematic model of a robotic system, which starts from $x = f(\theta)$, $\dot{x} = J\dot{\theta}$

So, from this, we can, from equation 2, which is the differential kinematics equation, we can say that $\dot{\theta} = J^{-1}\dot{x}$

So, now this is a situation called inverse kinematics. As I mentioned earlier in the lecture, inverse kinematics is for controlling a manipulator. Let us see that. Let us consider the control law for this system: x dot equals j theta dot for this non-linear system when we consider a control law so that the non-linear system becomes the closed-loop aerodynamic equation becomes linear and stable. We can say that the control law is a feedback control law and this approach is a feedback linearization approach.

So, let us choose a control law which is given by $\dot{\theta} = J^{-1}(K_p E + \dot{x}_d)$ So, this is the control law. Let us choose this control law. When we substitute back into this equation, we get

$\dot{x} = J\left(J^{-1}(K_p E + \dot{x}_d)\right)$

which leads to the first-order dynamic equation which is given by $\dot{E} + K_p E = 0$

where $E = x_d - x$ that means desired Cartesian position of the end effector and x is the actual Cartesian tip position of the robot.

Similarly, $\dot{E} = \dot{x}_d - \dot{x}$ upon substituting back because we know that $\dot{\theta} = J^{-1}\dot{x}$

that theta dot is given by the control law that is j inverse into x d dot plus k p into error so

Let us equate this to this equation which is $J^{-1}(\dot{x}_d + K_p E) = J^{-1}\dot{x}$

which is nothing but a first-order error dynamic equation.

which is $\dot{E} + K_p E = 0$ which implies that E tends to 0 as time tends to infinity.

So, depending on the positive values of K_P, we can ensure that E tends to 0. Now, let us talk about an n-degree-of-freedom robot manipulator. Whose dynamic equation is given by $\tau = M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta)$

where tau is the n x 1 joint torque vector and theta double dot, theta dot and theta are nothing but n x 1 joint angular vector, n x 1 joint angular velocity vector, and n x 1 joint angular acceleration vector. Joint angular acceleration vector, theta dot is joint velocity vector, and theta is joint angle vector, whose individual sizes are n x 1.

And what is M of theta? It is an n x n matrix called the mass matrix or inertia matrix, inertia matrix. And C of theta comma theta dot represents the Coriolis and centrifugal matrix, which is of size n x n. And G of theta is an n x 1 gravity vector. Now, as per equation 1, we have the system dynamics given by this equation, which is highly coupled for an n-degree-of-freedom robotic manipulator. Now, we choose a control law, choose a control law tau equal to alpha tau dash plus beta, as we have seen in the previous class model-based control strategy for robot

manipulators. We have chosen this control law $\tau = \alpha\tau' + \beta$

where $\alpha = M(\theta)$, $\beta = C(\theta,\dot{\theta})\dot{\theta} + G(\theta)$ and tau dash is nothing but the servo law,

which is given by
$$\tau' \Rightarrow \text{Servo law.} \Rightarrow \ddot{\theta}_d + k_p E + k_D \dot{E}$$

Again, tau dash is nothing but the servo law, which is given by

$$\tau' \Rightarrow \text{Servo law.} \Rightarrow \ddot{\theta}_d + k_p E + k_D \dot{E}$$

where
$$E = \theta_d - \theta$$
$$\dot{E} = \dot{\theta}_d - \dot{\theta}$$

Now, substitute the control law chosen tau equal to alpha tau dash plus beta with what is tau dash and alpha and beta, substituting back in this control law, we have one huge expression. Let that expression

With the values of tau dash, the expression of tau dash alpha and beta substituted back into the control law, let that equation be equation number 3. Now, substitute equation number 3 back into equation 1. Equation 3 in system dynamics 1,

$$\alpha(\ddot{\theta}_d + k_p E + k_D \dot{E}) + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) = M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta)$$

Here, beta is c of theta comma theta dot into theta dot plus g of theta.

This represents the beta value. This represents the tau dash value, okay? And then, alpha can be replaced by m of theta. So, this equation is going to turn out to be

$$\ddot{E} + k_D \dot{E} + k_P E = 0$$

That means we finally come out with the second-order error dynamic equation upon substituting the chosen control law back into the system dynamics. Thus, the closed-loop system dynamics becomes the second-order error dynamic equation, which is stable and linear.

So, with this, I come to the conclusion that In this lecture, we have seen the design of feedback linearization where the chosen control law u can be substituted back into the system dynamics in order to make the closed-loop system dynamics linear and stable. We have seen feedback linearization design, and then we have seen an example involving one degree of freedom single-link revolute jointed manipulator, one R manipulator precisely. It is a planar manipulator. We have seen applying the control law, and then we have seen kinematic control of a robot manipulator using feedback linearization with the control law being theta dot equal to J inverse into X d dot plus k_p into error.

This is basically the control law with the common and popular nomenclature CLIK, closed-loop inverse kinematics, which ensures that given a desired trajectory like this, your actual trajectory with the error between the desired trajectory and the actual trajectory in the Cartesian space can be taken into account by the control law given by CLIK, so that During the inverse kinematic process, you will have the desired trajectory here, and your actual trajectory will also be merged or tracking the x_d(t) more accurately. That accuracy is ensured by the term Kp, which is a positive definite matrix value for an n-degree-of-freedom robotic system. So, we have seen how feedback linearization is

implemented for a kinematic model-based robotic system control. Then, finally, we have seen the feedback linearization control approach for an n-degree-of-freedom robot arm, where it has n degrees of freedom, which means that the n links, you can say, and n joints. For such a system, we have used the control law tau equal to alpha tau dash plus beta. We have used this control law in order to have the closed-loop system dynamics being stable. So, as I mentioned, for the kinematic control, we had E_dot + Kp E = 0, a first-order aerodynamic equation. For the dynamic model-based control with the feedback linearization, we eventually ended up with a second-order

error dynamic equation. That is the difference between applying feedback linearization for kinematic model-based control of a robotic system and dynamic model-based control of a robotic system. So, with this, I stop here, and you people can try solving a two-degree-of-freedom robotic system, which is an R and R planar robot. You can try solving by computing the dynamic model.

If you have the dynamic model of the simplest two degrees of freedom revolute jointed robotic system which means X this is Y. If you have this one this is theta 1 theta 2 and this is L 1 L 2. If you have such a two link planar manipulators dynamic model you can try that by having the partition controller tau equal to alpha tau dash plus beta and try to have the trajectory for both theta 1 theta 2 matches perfectly that is all that is theta in degree time in second degree okay that's all i stop here thank you very much for your patience the next class we will see about light no stability theory followed by robot programming thank you very much