INTELLIGENT CONTROL OF ROBOTIC SYSTEMS

Prof. M. Felix Orlando

Department of Electrical Engineering

Indian Institute of Technology Roorkee

Lecture 12: Neural Adaptive Control of Robotics Systems

Good morning, everyone. Today, we are going to see neuro-adaptive control of robotic systems. The outline of this lecture will be as follows. First, we are going to see about filtered error approximation-based control. Then, we have adaptive control. Of robotic systems.

And we have. Neuro. Adaptive. Control. Of robotic systems.

So, let us talk about. Approximation-based. Based controllers. We all know that. Computer torque control, which is given by the control law $\tau = \alpha \tau' + \beta$

where tau dash is the servo law and alpha is $B(q)$ which is the inertia matrix, and

$$\beta = C(q,\dot{q})\dot{q} + h(q)$$

So, with this computer control scheme, it works very well provided all the dynamical parameters, such as B(q), $C(q,\dot{q})\dot{q}$ and $h(q)$ are known accurately.

Then only this computer control will work well. But in the case of robotic systems, These parameters vary with time and are also not properly known. So, the control objective is to achieve trajectory tracking under parameter

uncertainty. That is the control objective because, for the robotic system, these parameters vary with time or they are also not known accurately. So, the control objective for us is to achieve trajectory tracking task under parameter uncertainty. There are two approaches to deal with this: one is adaptive control, Another one is the robust control scheme. So, in this lecture, we are going to see adaptive control followed by neuroadaptive control. Then we will see what is filtered error. Because the topic we are going to cover in the first part is filtered error approximation-based controller.

First of all, filtered error. It is given by $r = \dot{e} + \Lambda e$

where e is the tracking error given

$$e \Rightarrow q_d - q$$

which is the error between the desired joint trajectory and the actual joint trajectory. That is called the tracking error. Tracking error. Whereas lambda is a positive definite design parameter matrix.

Lambda is a positive definite design parameter matrix. Now, taking the Laplace transform for a scalar case, we get $E(s) = \frac{R(s)}{s+\lambda}$ So that is giving you the error,

where the tracking error e can be obtained by passing this quantity R, which is a filtered error, through a low-pass filter, and hence the name. So again, the tracking error e can be obtained by passing this quantity R

through a low-pass filter, and hence the name. So this kind of error representation leads to a simplified controller design owing to the model reduction or the system order reduction. Next, coming to the boundedness of this error. Let us talk about the boundedness of the filtered error ensures the boundedness of the tracking error as well as the time derivative of the tracking error.

Thus, the tracking error norm is bounded by the $\|e\| \leq \frac{\|r\|}{\sigma_{min}(\Lambda)}$

and $\|\dot{e}\| \leq \|r\|$

Thus, the boundedness of R filtered error ensures the boundedness of both the tracking error as well as the derivative of the tracking error and also the desired trajectory. The desired trajectory is bounded such that

$$\left\| \begin{matrix} q_d(t) \\ \dot{q}_d(t) \\ \ddot{q}_d(t) \end{matrix} \right\| \leq q_B$$

Okay, so the boundedness of both the error as well as the desired trajectories is seen now.

Next, we will see how the dynamic model can be expressed in the filtered error. So, the expression of the dynamic model of the robotic system in terms of filtered error is given by $B\dot{r} = -C(q,\dot{q})r + f(x) - \tau$

That is the dynamic model of a robotic system expressed in the filtered error r, where the non-linear robot

$$f(x) = B(q)(\ddot{q}_d + \Lambda\dot{e}) + C(q,\dot{q})(\dot{q}_d + \Lambda e) + F(\dot{q}) + G(q)$$

and we have the state vector $x = \begin{bmatrix} e^T & \dot{e}^T & q_d^T & \dot{q}_d^T & \ddot{q}_d^T \end{bmatrix}^T$

And F of X contains all the potentially unknown robot parameters. That is f of x. So, it is a non-linear function. And it has all the unknown robot parameters.

Okay. So now we can move on to the continuation of approximation-based control. Now we are going to have the control law. That is called an approximation-based controller. Let us see that.

Approximation. Based controller is given by a control law $\tau = \hat{f} + k_v r - v(t)$

where f hat is an estimate of f of x, and v of t is the robustifying term of this control law,

and KVR is an outer PD tracking loop. So, outer PD control loop. Now, with this control law put in that equation, which is the dynamic equation in terms of the filtered error, we substitute back here in the previous equation, we get the closed-loop error dynamics,

$$B\dot{r} = -C(q,\dot{q})r - k_v r + \tilde{f} + v(t)$$

Now there is something called f tilde, which is nothing but $\tilde{f} = f - \hat{f}$

Where f is the actual function, and f hat is the estimated function. And f tilde is given by the term function approximation error.

Okay. So now let us move on to the controller design problem statement. Now let us talk about the controller design problem statement. Controller design problem. So we need to select a controller.

The estimate f hat and v(t), the robustifying term in the above approximation-based control law, such that the closed-loop error dynamics is linear and stable. Linear and stable, such that the closed-loop error dynamics is linear and stable. So, for that, we need to select the estimate f hat and robustifying term v of t to make the closed-loop error dynamics linear and stable. Now, let us talk about Adaptive control.

Let us talk about adaptive control. Adaptive control of robotic systems. It can be applied to a class of non-linear systems. Non-linear systems. That satisfy the following conditions.

Such as. The systems of full state. Is measurable. The non-linear. Then the second is.

The non-linear. Plant dynamics. Can be. Linearly parametrized. Second is, the non, second condition is the non-linear plant dynamics can be linearly parametrized.

Then, third one, the non-linearities can be canceled stably. The non-linearities can be canceled stably. Stably. By the control input. Control input.

Okay. If the parameters are known. If the. System parameters. Are.

Known. So, in this case. The adaptive control can be. Applied. So, now let us rewrite the robot dynamics again in terms of the filtered error.

We will be getting

$$B\dot{r} = -C(q,\dot{q})\,r + f(x) - \tau$$

The C matrix depends on both the generalized coordinate q as well as the derivative of it. So, I simplified that here without putting the input arguments of the C matrix. So, where the nonlinear function here, the nonlinear function is f (x), may be expressed as a may be expressed as a linear function of unknown parameters

$$f(x) = D(q)(\ddot{q}_d + \Lambda\dot{e}) + C(q,\dot{q})(\dot{q}_d + \Lambda e) + F(\dot{q}) + G(q) = W(x)\,\phi$$

where w(x) is known and it is called it is called the regression matrix matrix and phi is a function of or is a vector Is a vector. Of unknown parameters. Unknown parameters.

Okay. So, this is the situation f (x) can be represented by w(x) into phi. Where w(x) is a known regression matrix and phi is a vector of unknowns.

One adaptive control law proposed by Slotine in the year 1988 is given by

$$\tau = W(x)\,\hat{\phi} + K_v r$$

Which is equal to $\hat{f}(x) + K_v r$

And we need to know about phi hat. So, phi hat is obtained by $\dot{\hat{\phi}} = \Gamma W^T(x)\, r$

where gamma greater than 0 is a tuning function. Parameter. Matrix. And we will be.

Getting this. Expression for phi hat dot. In terms of gamma W transpose.

So the control law proposed by Slotin is $\tau = W(x)\,\hat{\phi} + K_v r$

Where phi hat is obtained by the update law, which is given by phi hat dot equal to gamma, which is a tuning parameter matrix W transpose of X into R.

And this expression, say expression A, is obtained by Lyapunov stability analysis that we are going to see now.

So, now the estimate of non-linear function is given by $\hat{f}(x) = W(x)\hat{\phi}$ and due to the linear in the parameter assumption (LIP assumption).

We can have $\tilde{f} = f - \hat{f} = W(x)\phi - W(x)\hat{\phi}$
$\Rightarrow W(x)\tilde{\phi}$

where w(x) is a regression matrix and phi tilde is given by $\phi - \hat{\phi}$

and f tilde is given by the function approximation error which is given by f minus f hat. Okay, so now let us continue. Now, the above adaptive control is following closed-loop error dynamics. The above adaptive controller yields the following closed-loop error dynamics in terms of filtered error. That is $B\dot{r} = -C(q,\dot{q})r - k_v r + W(x)\tilde{\phi}$

Now let us select the Lyapunov function candidate. Function candidate. That is given by

$$L = \underline{\frac{1}{2} r^T B(q) r} + \frac{1}{2}\tilde{\phi}^T \underline{\Gamma^{-1}\tilde{\phi}}$$

One is the first term of this Lyapunov function candidate is basically the error term which is a filtered error term and another one is the function approximation error term. It is the parameter estimation error term in the second term of Lyapunov function candidate and the first term is filtered error term. Let us continue this. So, here we can say that the lambda is a symmetric positive definite weighting matrix where I can write here where lambda this lambda is

A symmetric positive definite weighting matrix. That is all. Now, let us go for the time derivative of the Lyapunov function candidate that is given by

$$\dot{L} = \frac{1}{2} r^T \dot{B} r + r^T \underline{B\dot{r}} + \underline{\tilde{\phi}^T \Gamma^{-1}\dot{\tilde{\phi}}}$$

Now, substituting the closed-loop error dynamics B R dot, we will be getting this L dot being

$$\dot{L} = \frac{1}{2} r^T \dot{B} r + r^T\left[-C(q,\dot{q})r - k_v r + W(x)\tilde{\phi}\right] + \left[\tilde{\phi}^T \Gamma^{-1}\dot{\tilde{\phi}}\right]$$

So, that leads to $\cancel{\frac{1}{2} r^T (\dot{B} - 2C) r} - r^T k_v r + \tilde{\phi}^T \left(\Gamma^{-1} \dot{\tilde{\phi}} + W^T(x) r \right)$

That can be given by having half R transpose taken out, we will get V dot minus 2C into R minus R transpose KVR. Plus phi tilde transpose taken out, we have gamma inverse phi tilde dot plus W transpose of X into R. Now we can see that we have arranged this equation to this equation so that we can have the skew symmetric.

Matrix $\underline{\dot{B} - 2C}$ as per the properties of the dynamics of a robotic manipulator. Based on the skew-symmetric property of this matrix $\underline{\dot{B} - 2C}$

we can have $x^T (\dot{B} - 2C) x = 0$ for all x.

So, we are going to have this first term on the right-hand side being 0. So, we will have this $\dot{L} = \underline{-r^T k_v r} \leq \underline{0}$

This is obtained. And now we have to check one thing: that the L dot transpose is obtained by the L dot. That is, the derivative is obtained as negative definite. And in this case, we are left with only this term. Because we are having the second term, which is evaluated by $\dot{\hat{\phi}} = -\dot{\tilde{\phi}} = -\Gamma W^T(x) r$

That's why $\dot{\tilde{\phi}} = -\dot{\hat{\phi}}$ which is nothing but phi minus phi hat dot.

So, phi dot is zero because phi is a constant, and hence the inner bracket of the previous equation becomes $-\Gamma^{-1} \dot{\hat{\phi}} + \underline{W^T(x) r = 0}$ in the previous equation of L dot.

So, this term must be 0, for which $\dot{\hat{\phi}} = \Gamma W^T(x) r$

That is what is given along with Slotine's control law for the update of phi hat. Okay.

So, now, substituting this back into the L dot equation, we finally end up with this equation, say, B. That is, $\dot{L} = -r^T k_v r \leq 0$

And therefore, r & $\tilde{\phi}$ are bounded according to Lyapunov's theorem. According to Lyapunov's theorem. Okay. So, now we have to ensure that r(t) tends to 0 as time tends to infinity.

We have to ensure r(t) tends to 0 as t tends to infinity. So, one may use Barbalat's lemma to show that $\dot{L} \to 0$ with time. So let us take the double derivative of L. That means the derivative one more time of L dot. So we get $\ddot{L} = -2r^T k_v \dot{r}$

And substitute the error dynamics here. So we get in terms of B inverse. So L dot with the substitution of substitute error dynamics, closed loop error dynamics. Okay, so we get

$$\ddot{L} = -2r^T k_v B^{-1}\left[-Cr - k_v r + W(x)\tilde{\phi}\right]$$

Now, the right-hand side is bounded. Now, in the previous equation, say C, the right-hand side is bounded because of the boundary conditions of the manipulator parameters such as the C matrix, B matrix, etc. And the demonstrated boundedness of r & $\tilde{\phi}$

Thus, I can say that the right-hand side is bounded because of the bounded conditions. Assumptions of the system parameters of the robotic system, which are the B matrix, C matrix, etc.

And also, previously, we demonstrated that r, the filtered error, and phi tilde, the parameter estimation error, are bounded. And hence, we can say that L double dot is bounded, which implies that L dot is bounded. Uniformly continuous, and by Barbalat's lemma, we can say that L dot tends to 0 as t tends to infinity. Thus, one can say that R of t tends to 0 as t tends to infinity. That is all.

Now, we move on to the neuroadaptive controller. That means adaptive control based on neural networks. Let us see that. That is a neural network-based adaptive control. This is the major portion of today's lecture.

Again, let us say or let us express the dynamic model of the robotic system in terms of the filtered error r. So, it becomes

$$B\dot{r} = -Cr + f(x) - \tau'$$

Now, choose the control law as $\tau = \hat{f} + k_v r$

Or where $\hat{f} = \hat{W}^T \phi(x)$ with f hat being an estimate of f where f I represented basically f(x). Let f(x) be represented by f hereafter. So f hat is an estimate of the function f(x) which is capital F which is basically a function of non-linear parameters. So here in neuroadaptive control, a neural network is used to approximate the this non-linear function, function f. So, the control law can be written again as

$$\tau = \hat{W}^T \phi(x) + k_v r$$

okay. So, now the closed-loop error dynamics is going to become with this control law, And the dynamic model, the closed-loop error dynamics is going to become

$$B\dot{r} = -Cr + F - \hat{W}^T \phi(x) - k_v r$$

which implies $$B\dot{r} = -Cr + \underline{W^T \phi(x)} - k_v r - \hat{W}^T \phi(x)$$

Since $F(x) = W^T \phi(x)$ Simply we are representing $F(x) = W^T \phi(x)$

Therefore $$B\dot{r} = -Cr - k_v r + \tilde{W}^T \phi(x)$$

which is say equation number 1, where W tilde is given by W minus W hat. So, now consider the Lyapunov function candidate $L = \frac{1}{2} r^T B r + \frac{1}{2} tr\left[\tilde{W}^T \Gamma^{-1} \tilde{W}\right]$

Therefore, time derivative of this Lyapunov function candidate $\dot{L} = r^T B\dot{r} + \frac{1}{2} r^T \dot{B} r + tr\left[\tilde{W}^T \Gamma^{-1} \dot{\tilde{W}}\right]$

which implies R L dot equal to R transpose substituting the closed-loop error dynamics in this here. Equation L dot. So, we get R transpose equal to this L dot equal to

$$r^T\left[-Cr - k_v r + \tilde{W}^T \phi(x)\right] + \frac{1}{2} r^T \dot{B} r + tr\left[\tilde{W}^T \Gamma^{-1} \dot{\tilde{W}}\right]$$

So, now we are just expanding the equation of L dot, and then in this equation again we have C matrix, B matrix. So, B dot and C. So, as per again the skew symmetric matrix property of the skew symmetric matrix, $(\dot{B} - 2C) \Rightarrow r^T(\dot{B} - 2C)r = 0$

in the previous equation's expansion and from the properties from the property of trace or matrix trace, we can say that

$$r^T \tilde{W}^T \phi(x) = \text{trace}\left(\tilde{W}^T \phi(x)\, r^T\right)$$

that is, row matrix multiplied by column row matrix multiplied by a column matrix can be written in terms of trace as trace of column matrix multiplied by row matrix. So, with this property of matrix trace, we can represent R transpose W tilde transpose phi of X equal to trace of W tilde transpose phi of X into R transpose. Thus,

$$\dot{L} = -r^T k_v r + tr\left[\tilde{W}^T \phi(x) r^T\right] + tr\left[\tilde{W}^T \Gamma^{-1} \dot{\tilde{W}}\right]$$

Which finally leads

$$\dot{L} = -r^T k_v r + tr\left[\tilde{W}^T \phi(x) r^T + \tilde{W}^T \Gamma^{-1} \dot{\tilde{W}}\right]$$

where W is constant. So, W tilde dot is going to become minus W hat dot. So, we are going to have the expression for

$$\dot{L} = -r^T k_v r + tr\left[\tilde{W}^T\left[\phi(x) r^T - \Gamma^{-1}\dot{\hat{W}}\right]\right]$$

Since $\dot{\tilde{W}} = -\dot{\hat{W}}$ This is it. So, this middle term $\phi(x) r^T - \Gamma^{-1}\dot{\hat{W}} = 0$

So, to have this entire term 0. So, this term to be 0 means you are going to get $\dot{\hat{W}} = \Gamma \phi(x)\, r^T$

that is the thing. So, in that With this expression of W hat dot, $\dot{L} = -r^T k_v r \leq 0$

Now, $\ddot{L} = -2\, r^T k_v \dot{r}$ and we can say that L dot is going to become 0.

With t tends to infinity, and finally by Barbalat's lemma, as in the case of adaptive control, we can say finally that r(t) vanishes, okay, because of the boundedness of the system parameters. That is here we involve B inverse of Q, with the boundedness of B inverse of Q and all the signals associated with the closed-loop error dynamics, it can be verified that R and that is R dot and L double dot are bounded. That leads to R tends to 0 as time tends to infinity. So, with this, I conclude that in this So, coming to the conclusion, we can say that in this class we have seen filtered error approximation-based control of robots, precisely adaptive control of robotic systems, and thirdly, neuroadaptive control of robotic systems. So, in the neuroadaptive, we had W that is obtained from the weight matrix of the neural network, weight matrix of neural network, and phi of X is basically the transfer function of the neuron or the activation function associated with the neurons of the neural architecture. Thank you so much. for your time.