Advanced Aircraft Control Systems With MATLAB / Simulink

Prof. Prabhjeet Singh

Department of Aerospace Engineering

Indian Institute of Technology Kanpur

Lecture 58

MATLAB and SIMULINK implementation of complete 6 DOF aircraft dynamics

So welcome friends. In the last lecture, we have developed all the necessary tools which is required for running the 6 DOF aircraft equations of motion. So without wasting any further time, let us quickly switch back to MATLAB. So I have written the code already here. First, let us run the six DOF simulation directly in MATLAB. So here the initial condition will remain same. That is 30 into cos t. V cos α. So, here α is assumed to be initial angle as 2.1471 degrees and side velocity I am considering it as 0 and the downward vertical component velocities we calculated again it was V sin α your total V is 30 meters per second sin into α. And the rates I'm assuming initial conditions to be zero. Similarly, the lateral parameters, ϕ naught, I'm considering it as zero. That is roll angle, initial roll angle. an initial yaw angle and θ naught remains same that is the α equals to initial α naught equals to 2.1471 degrees and initial position x naught y naught and z naught also i am considering the initial values as 0. now i am putting this initial conditions in a vector in this format like we did for 3 DOF simulation. So, initial conditions we have $u_0, v_0, w_0, p_0, q, r_0, \phi_0, \theta_0, \psi_0, x_0, y_0, z_0$.

So, we have a total of 12 states. All right. The time span I am considering is the same, 0 to 300 seconds, and the ODE solver I am using is also the same, which is ODE45, that uses Runge-Kutta. The output is t and the states. Here, we have 12 states instead of 6, which was there in the 3-DOF simulation. The syntax for ODE45 remains the same, meaning we call the function file 'equations of motion,' the time span (0 to 300 seconds), with the given initial conditions that we have over here. And when it comes to line number 24, it looks for 'equations of motion. And I have included it directly in this MATLAB file, that is, 'dy/dt equals equations of motion. Again, this is a tilde. You can also include it as 'time.' And y is a vector of 12 states. And the output file I have named as 'output variable.'

I have named it as 'dy/dt.' And these are the parameters that remain the same. So, I have not written the lateral parameters. Let me write the lateral parameters first. So that it will be easier for you to simulate by yourself. So here we have

$$C_{y0} = C_{l0} = C_{n0} = 0$$

$$C_{y\beta} = -0.98, \quad C_{l\beta} = -0.12, \quad C_{n\beta} = 0.25$$

$$C_{yp} = 0, \quad C_{lp} = -0.26, \quad C_{np} = 0.022$$

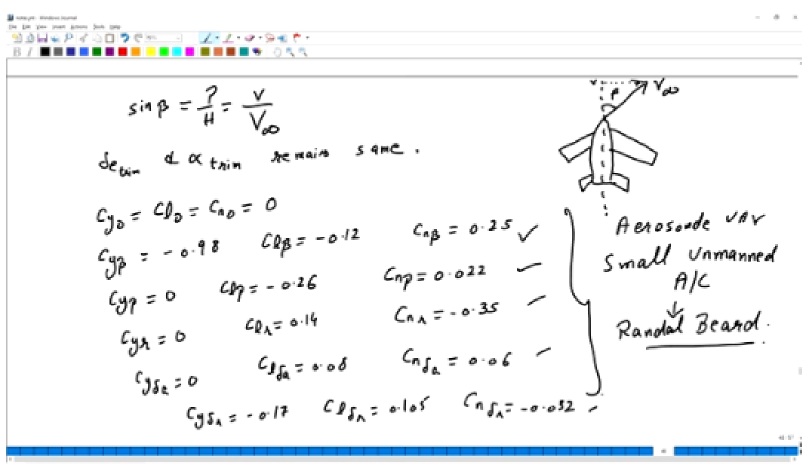
$$C_{yr} = 0, \quad C_{lr} = 0.14, \quad C_{nr} = -0.35$$

$$C_{y\delta a} = 0, \quad C_{l\delta a} = 0.08, \quad C_{n\delta a} = 0.06$$

$$C_{y\delta r} = 0, \quad C_{l\delta r} = 0.105 \quad C_{n\delta r} = -0.032$$

Again, these parameters belong to the Aerosonde UAV, which I have considered from a book by Randal Beard, Small Unmanned Aircraft. And these derivatives, they belong to a particular equilibrium point, particular trim value. Please note that to get these derivatives, it is actually not easy. A lot of effort is put to find these values. And there are multiple ways to get these values. Of course, one is the wind tunnel test.

(Refer Slide Time: 05:47)



Multiple wind tunnel tests are done to get the approψmate of these derivatives. The second way you can find is through CFD simulation. The third method is the linearization technique or small perturbation theory. which we have already covered in the first part of this course. Another method to obtain these derivatives is flight testing. So after flight testing, we can use system identification techniques. We can use system identification techniques to find these derivatives. So the point I want to stress here is that it is not easy to get these derivatives. All right. Now let us go back to MATLAB. These are the initial parameters, then product of inertia, gravity I am considering 9.81, sea level density 1.225, span, again the thrust remains the same, 2 kgs of thrust multiplied by 9.81 to get in newtons. We know the thrust should be in newtons. And these are the aerodynamic coefficients. Which I've already mentioned before. In case I missed any coefficient, you can note it from here. And these are the inertia expressions: $\Gamma, \Gamma_1, \Gamma_2$and so on and so forth.

Now, finally, I'm letting MATLAB know that these are the states at u, v, w velocity is in the body frame, then we have angular rates p, q, and r. Since there are 12 states, I'm letting MATLAB know that these are the states: ϕ is the seventh state, θ, ψ, similarly x, y, and z. So this is basically unused in the equations, but that's all right. So, in 3 DOF, we had v equals to the square root of u squared plus w squared only, but in 6 DOF, we will be having v equals to the square root of u squared plus v squared plus w squared, that is the total velocity magnitude. And we only added α equals to tan inverse of w upon u. Here we have β also, A sin small v upon capital V. All right. Then we have aerodynamic forces and moments. I have modeled here

$$C_L = C_{L0} + C_{L\alpha}\alpha + C_{L\delta e}\delta_e$$

$$C_D = C_{D0} + C_{D\alpha}\alpha$$

You can model it as

$$C_D = C_{D0} + KC_L^2$$

also if those parameters are given. And the side force coefficient

$$C_y = C_{y0} + C_{y\beta}\beta + C_{yp}\frac{pb}{2V} + C_{yr}\frac{rb}{2V} + C_{y\delta a}\delta_a + C_{y\delta r}\delta_r$$

Then the rolling, pitching and yawing moment coefficients are given in this format that we have already discussed. For example,

$$C_l = C_{l0} + C_{l\beta}\beta + C_{lp}\frac{pb}{2V} + C_{lr}\frac{rb}{2V} + C_{l\delta a}\delta_a + C_{l\delta r}\delta_r$$

$$C_m = C_{m0} + C_{m\alpha}\alpha + C_{m\delta e}\delta_e$$

And finally, finding the lift, drag and the side force. C_L we have already found it here.

Similarly, C_D we found in this 75th line code. C_y we have found in 76th line code. All right. Next, we have moments. Moments about x-axis, moment about y-axis, moment about z-axis, Similarly, C_m and C_l. The throttle is given as $T_{max}\delta_t$. δ_t, we have assumed it as 0.5. That is 50% of throttle is always applied during the entire simulation run. And side force is simply side force coefficient that is not coefficient forces in y direction. This is the side force. Hence, we had derived this equation earlier. And we can write even in the terms of accelerations in x direction, y direction, z direction, the process remains same as of three degree of freedom equations of motion. So, fx by m minus g into sine θ, that is linear acceleration along x axis. Similarly, acceleration about y axis and linear acceleration along z axis and finally so this is $\dot{u}, \dot{v}, \dot{w}$ then we have accelerations $\dot{p}, \dot{q}, \dot{r}$ that we have derived in today's on the last week's lecture Then we have $\dot{\phi}, \dot{\theta}, \dot{\psi}$ expressions, kinematic equations. And finally, the position derivatives, $\dot{x}, \dot{y}, \dot{z}$.

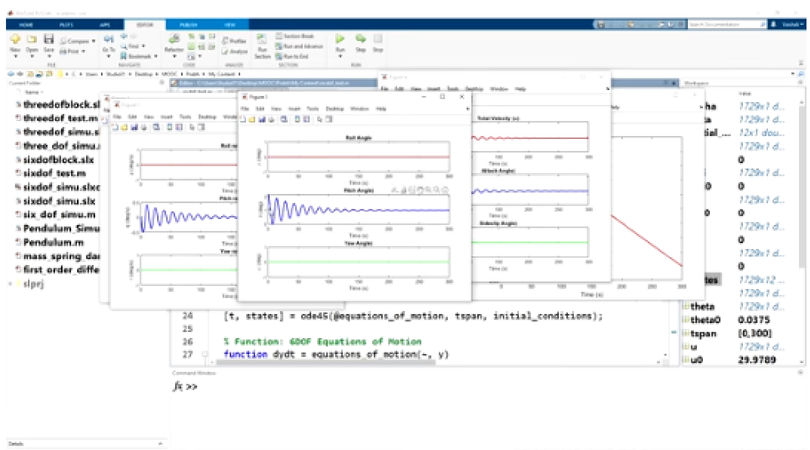
And the output vector is dy dt, which I have included here: dy dt. So, I'm not taking any other output here. I'm interested only in the 12 states. Hence, here: $\dot{u}, \dot{v}, \dot{w}, \dot{p}, \dot{q}, \dot{r}, \dot{\phi}, \dot{\theta}, \dot{\psi}\dot{x}, \dot{y}, \dot{z},$. Now, after this, when the MATLAB code reaches the 127th line, after it comes out of the function file, we need MATLAB to know that the first column belongs to u, the second column belongs to v, and so on. So, we have 12 states. So, if I run this till here, let me run it. So, we have states. We have 12 states over here. If I double-click over here, we get these 12 states. And we are letting MATLAB know which column is what. All right. I am removing this break line code. Next, we have to calculate V, α, and β again.

Since u is a vector now, we have to do element-wise multiplication. Hence, I put a dot here; otherwise, it will throw an error. Similarly, α is atan inverse of w divided by u, and asin v upon capital V. Now, finally, we are ready for plotting the results. So, in Figure 1, I have included plotted subplots: t with u, then the second plot is with side velocity small, and the third is the downward component of velocity w. So, Figure 1 indicates velocity components in the body axis. Figure 2 indicates the angular rates, where we have P converted into degrees—that is, radians to degrees multiplied by 180 divided by pi. Then, the second plot is pitch rate q, and the third plot is r. And Figure 3 indicates plots of Euler angles, that is, ϕ, θ, and ψ. These also I have converted into degrees.

And finally, in Figure 4, I am plotting the total velocity V separately, which we have calculated here in line number 133. Similarly, I plotted α and β here. And in Figure 5, I am only interested in knowing the altitude, and I put a minus sign here. Since the z-direction is pointing toward the center of the Earth, we would like to know whether the altitude is increasing or decreasing with respect to the body axis. Hence, I put a minus sign here.

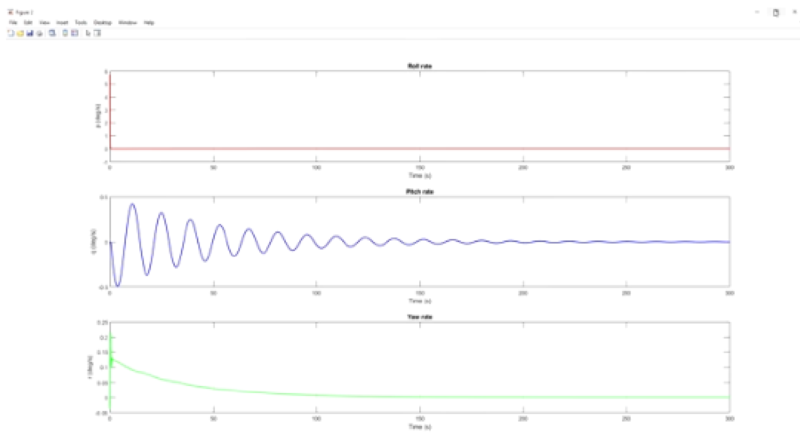
So, now let me continue running this code, and let us look at the plots. So, we will look at the plots one by one. First are u, v, and w. So, we know that we have already applied the trim condition, which is delta e trim and α trim. So, here the forward velocity is stabilized to around approψmately 30 meters per second, and since there is no perturbation or anything towards the body y-axis, this remains zero. And we have a small w over here, which is approψmately 1.1 meters per second. Now, moving to plot number two. So, there is no deviation in the lateral plots. So, there will be no deviation here. So, we get again the pitch rate approψmately equal to zero. And in figure 3 again, there will be no deviation in ϕ and ψ. There will be deviation or oscillation in θ, which is stabilizing to approψmately 1 degree. And in figure 4, I have directly plotted the total velocity. Here, we have the total velocity again at approψmately 30 meters per second. A is 2.14 degrees, and β is 0. And here again, we observe that the

(Refer Slide Time: 16:04)



The altitude is actually decreasing. Since there are no variables in lateral, it is behaving as if it is a three-degree-of-freedom motion. All right. Now, let me introduce a disturbance in the lateral variable. So, it is mostly affected by the angular rates. Let me set it as 0.1, which is radians per second. Now, if I run this code, Let us see the plots. Now, we have a small deviation in the side velocity V. Let us check P and R. Since I have given the deviation in P, and we know that lateral and directional dynamics are always coupled with each other. If there is a deviation in P, then there will be a deviation in R as well, which we can observe here. Although it is coming to zero. Since we have the α trim we applied and the delta E trim we applied. And ϕ reaches approψmately 0.4 degrees and then goes back to zero for the wings-level condition. And ψ—we actually don't care in which direction the aircraft is heading. θ is actually important. θ is approψmately around 1 degree here. Now, the velocity is approψmately 30 meters per second, and α is again approψmately 2.14 degrees.

(Refer Slide Time: 17:31)



And β also goes to 0. Now, likewise, I said earlier that α and W have a 1-to-1 relation. So, likewise, β and small v have a 1-to-1 relation. If there is small v, then there will also be β. Let me open the v plot. So, if there is a small v component, then there will always be a β component. If there is β, then there will be a small v component. So, there is always a 1-to-1 relation between these two. And finally, going to Figure 5. So, here again, we observe that the altitude is decreasing by 200 meters. Now, let us go a step further. Let me bring this back to 0. And we know that for longitudinal stability, $C_{m\alpha}$ Cm α should always be less than 0. So, here we have $C_{m\alpha}$; here it is negative. For longitudinal stability, we have Cm α should be negative. Let us see what happens when $C_{m\alpha}$ is not less than 0.

Now, I will try to run this simulation. So although the simulation has run, let us see the plots: ϕ, θ, and ψ. So here, θ is going up to 5 into 10 to the power 4, which is unrealistic. That means it is continuously doing a somersault, which is definitely not acceptable. So as soon as I put $C_{m\alpha}$ back to negative, then we have proper results. θ is back to its normal one degree. All right. So let me write a note over here. So for longitudinal static stability, $C_{m\alpha}$ should be less than 0. For a positive angle of attack, the pitching moment must decrease to bring the nose down and restore equilibrium. So when we have $C_{m\alpha}$ less than 0, the aircraft resists deviations from its trimmed angle of attack, maintaining stability in pitch. So here, $C_{m\alpha}$ is generally called the longitudinal static stability derivative.

Static stability derivative. Now, let us look at another variable. So, here we have the same α negative, all right. Now, let us look at C_{lp}. Here, $C_{lp}$was negative. Let me keep it as positive. Now, run the simulation. So, here, we still have good results. Why? Because we have not introduced any deviation in the lateral variable.

So, let me put again p here, p_0 as 0.1. And I'm running the simulation. It is running, still running. If you see the bottom left, it is showing as busy. It is still running. So, what it means is the solution is not able to converge to a finite value. So ultimately, there is no solution for this. That means the aircraft is no longer at an equilibrium point. I am stopping this simulation and then checking it with C_{lp} , changing it to minus. Let me put it as minus over here and run the simulation with the disturbance in the lateral-directional coefficients. So here, we still observe The aircraft returns to its equilibrium position in a finite time. All right. So, that is the significance of C_{lp}.

Now, what is C_{lp} ? Let me write it down. So, C_{lp} is nothing but the roll damping derivative. So, for positive p, for positive roll rate—what do you mean by positive roll rate? That is the right wing going down when viewed from behind. So, the right wing is going down, which indicates positive p. So, we need to generate a negative rolling moment. That means the left wing should go down. So, a negative rolling moment must be generated. That is, C_{lp} less than zero. So, what we did in the simulation was we kept C_{lp} as positive.

So that means once the right wing went down, so if this is positive, it further went down. So it should have an initial tendency to come back to its original position. So, hence, C_{lp} should be negative. Then finally, let us go back to another variable, which is C_{nr} . This is also negative. Let me keep it as positive over here, and we have an initial disturbance in the roll rate. I am running the simulation now. It is running. Showing busy here. So again, we are not able to get a converged solution, all right.

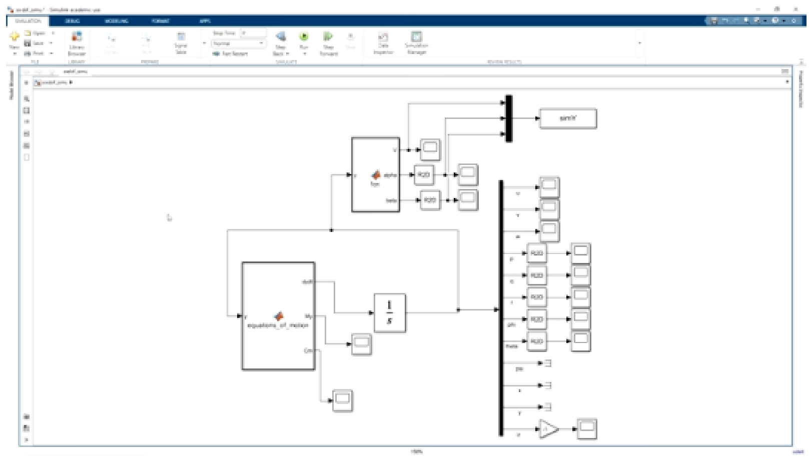
So, I am stopping it here. You can try it on your own. So, let me put it back to minus. Now, what is C_{nr} ? C_{nr} is Yaw damping derivative. Alright. So, for a positive yaw rate, that is for positive R, what do you mean by positive r? Right wing going backward when viewed from behind. So, when we have a positive yaw rate r, then a negative yawing moment must be generated. That is, the left wing should go back in order to be stable. So, that is, a negative yawing moment must be generated. That is, C_{nr} should be less than zero. And when we tried CNR greater than zero, with the initial disturbance, the right wing was already behind.

So then what happened? It kept on going backward. That means it kept on yawing. But for stability, we need it to return to its equilibrium position. So, hence, C_{nr} should be negative. All right. Now, let us write a note here. So, negative $C_{m\alpha}$, C_{lp}, and C_{nr} are part of the system's natural dynamics and restoring forces, ensuring the eigenvalues of the system dynamics have negative real parts, which is the requirement for stability. Alright. The $C_{m\alpha}$ significantly influences the short-period and phugoid mode. If you are not sure what the short-period and phugoid mode are, please refer to the first series of this course. C_{lp} and C_{nr} impact the roll subsidence. That is one.

It also impacts Dutch roll. And spiral mode, all right. So, similarly, we have $C_{l\beta}$ should be less than 0. This is for roll static stability derivative. And we have $C_{n\beta}$ greater than 0. This is for your static stability derivative. I am not going to test this now. I am leaving that as an exercise for you. Next, we have $C_{m\delta e}$, $C_{l\delta a}$ and $C_{n\delta r}$, so we call these primary control derivatives, followed by $C_{l\delta r}$ and $C_{n\delta a}$. These are usually pretty small values, so these are cross-control derivatives. So, control derivatives, that is both primary and cross-control. So, control derivatives can be thought of as gains. The larger the value of the control derivative, the larger the magnitude of the moment produced for a given deflection of the control surface. Now, let me go back to MATLAB again. So, I want you to, as an exercise, change these parameters from plus to negative and vice versa, and see how the model behaves. And also we have C_{m0} as negative. So we know that for trim, C_{m0} should be positive so that we can trim the aircraft at positive angles of attack. But we have C_{m0} here as negative. That's all right. Because we have already taken the α trim condition, we have elevator in the negative direction.

That means it is continuously pitching up. It is trimming the aircraft at positive angle of attack. All right. So I want you to change these parameters and see how the aircraft is behaving one by one. So that I leave that as an exercise to you. So, now we are done the simulation of MATLAB, simulation of aircraft equations of motion in MATLAB. Now, let us include Simulink as well. So, this is the Simulink model that I am trying to call from here, sim six DOF underscore sim dot slx. So, initial conditions remain same, u0, v0, w0 and so on and so forth. This is the initial vector. And this is the SLX model. Let me go to the Simulink model first. All right. Let me just delete it. OK.

(Refer Slide Time: 36:17)



So I'm not going to build this model again from scratch. So you can look at it. I'll open this MATLAB function file. All right. So here is the MATLAB function file. Everything remains same as was in the MATLAB.m file. Here output is of course the 12 state vector. I have also included moment about y axis. Similarly coefficient of moment cm about y axis and this is the function file equations of motion and everything remains same over here and this is the output vector u dot v dot w dot and so on and so forth. This is the function file. Here, I am actually integrating this. With the initial conditions as y_0 . All right. If I go to the model settings, model settings. I've used a fixed step here, 0.01 ODE45.

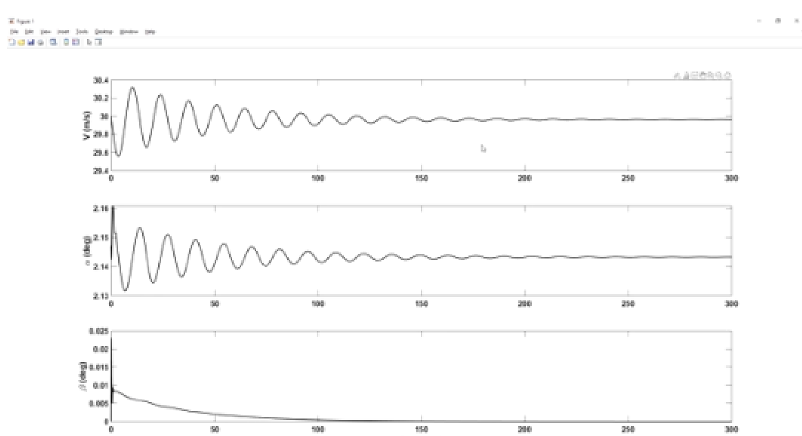
You can also use a variable step. It's up to you. Stop time is TF, as I've mentioned in the MATLAB code. Here, I'm extracting all the variables: u, v, w, p, q, r, ϕ, θ, ψ, x, y, and z. This symbol indicates the terminator. That means we are not plotting this. We are interested in these parameters. In z, I have multiplied this by minus one to track whether the altitude is increasing or decreasing. I have also transformed these variables to the workspace model. This is the workspace to workspace, as we did in the 3DOF simulation model. And I have extracted V, α, and β. What I have done here is I have added one more MATLAB function here. And in that MATLAB function, the only inputs required were u, v, and w. I calculated V, α, and β separately. So it is up to you. You can extract directly from this model as well. By writing here V, α, and β.

So here the output we get is V, total velocity, attack angle, and the side slip angle. All right. And here I have converted radians to degrees. And this is to workspace, as I said before. Now, let me run this code directly. Let me save it first. All right. Now, after this, MATLAB reaches the 11th line. It runs this Simulink model. And then I'm extracting the data, V, α, and β from there.

The workspace file that I created is the same Y, and then I'm plotting here. In Figure 1, I plotted V, α, and β. V, α, and β. So here, α I have already converted to degrees. I have already converted to degrees. Hence, I have not written here α multiplied by 180 over pi. So, this is in the subplot: we have T, V, then α and β. So now, we are ready to run this code. Let us see if we are getting similar results or not. Yes. So indeed, we are getting similar results. So, the total velocity is around 30 meters per second. The attack angle is approψmately 2.14 degrees. The sideslip is 0. So, let us see the other plots. This is θ. Θ should be around 1 degrees. Yeah, it is around 1 degrees. Since there is no perturbation in the lateral variables, this ϕ r should all be around 0. Similarly, p should be 0. Yes, it is. And q should go towards 0 at a certain time of interval. Yes, it is going to 0. This is a forward velocity which should be around 30 meters per second. Side velocity should be 0 and small w should be around 1 or 2 meters per second.

Yeah, it is around 1 meters per second. and this is the total moment about the y axis which is going to zero after certain time this is the coefficient of moment and the altitude it is actually decreasing all right and this is the total velocity v which should be around 30 meters per second α should be around 2 degrees yes it is 2 degrees and β should be around 0. All right. Let me change the initial conditions again and put p_0 as 0.1. You can put perturbations or deviations in other parameters as well. It is up to you. So, let me rerun the code. So, here we observe again perturbation in one lateral variable will instigate perturbation in another variable that is now here it is the perturbation is in side slip angle β. V is again still going towards 30 degrees sorry 30 meters per second and α is around 2.14 degrees.

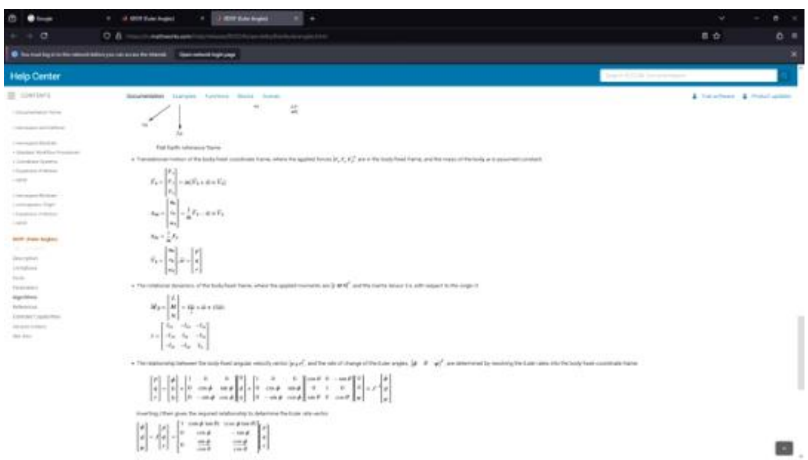
(Refer Slide Time: 42:05)



Let me go to the MATLAB in Simulink and rerun the code and rerun this scope and we get the similar results. θ is approψmately 1 degrees. Yes it is. Similarly velocity it is approψmately 30 meters per second and the altitude is actually again decreasing. What I also want you to do is Change this delta E that we have given the trim angle of trim elevator deflection changes to any arbitrary value and see now then what happens. So this I leave this an exercise to you. Similarly, you can change this variable delta A delta R and observe what is actually happening apart from changing these variables as well. All right. So now we have done simulation for aircraft six DOF both in MATLAB and in Simulink.

All right. Now, the one last thing left is the Simulink six DOF block that we can get from here. Six DOF block. Now, let me go back to the equations, the six DOF equations that we have seen earlier. All right. So, we see over here that m into VB dot plus omega, omega is PQR cross total velocity V. So, let me write $\dot{p}, \dot{u}, \dot{r}, \dot{v}, \dot{w}$. Let me write the omega dot first from this expression. Yeah, here it is, omega dot. Let me write omega dot. We have I inverse mp moments minus omega cross I omega.

(Refer Slide Time: 43:51)



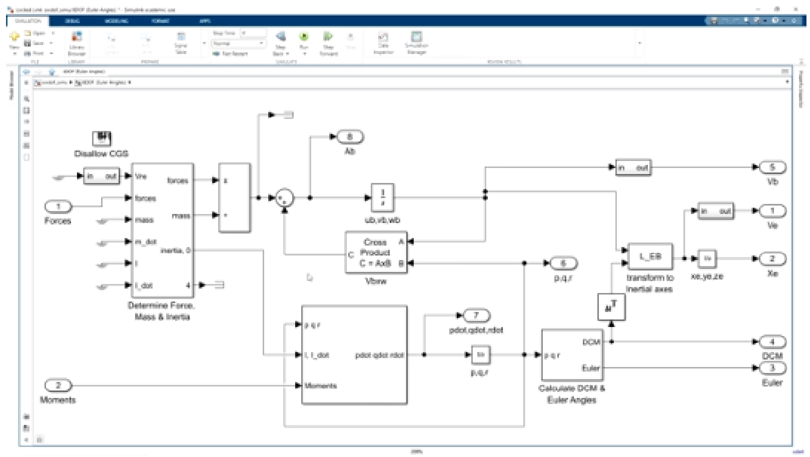
Similarly, we have velocities $\dot{u}, \dot{v}, \dot{w}$ in this format, 1 by m fb. So, whatever equations we had written before, we simplified these equations in that format, and we get those equations. Now, this is in matrix form. Equations are the same, only the form is different. All right. And similarly, we had the θ dot equation as well. Let me write only this equation, that is the $\dot{\theta}$. $\dot{\theta}$ can be written as

$$\dot{\theta} = q\cos\phi - r\sin\phi$$

Now, let us go back to the Simulink model and see here. Likewise, in the 3 DOF model, we have here the inputs as forces. Only this time, this is in a vector form: fx, fy, and fz, and the moments are in mx, my about the x, y, and z axes. So, this I am leaving as an exercise for you. So, all you have to do is put the forces vector here, similarly the moments vector here, like we did for the three equations of motion. We have output here: velocity, ϕ, θ, ψ, DCM (the direction cosine matrix), velocity in body axis rates PQR, and then accelerations in the body axis.

Now, when we double-click this, what we get over here, the units are in matrix. Mass is actually fixed; we are not changing the mass. It needs input as x_e, y_e, and z_e. Similarly, initial velocity in body axis is zero, zero, zero. Similarly, orientation angles ϕ, θ, and ψ, initial body rotations p, q, r, and initial mass. So, one thing we also had—let me go back to the 3 DOF model. So, in the earlier lecture, we used this model. If you double-click this, then we enter the parameters over here as well. The initial velocity is 100 meters per second. The initial body attitude is θ_0. We need to enter it over here. This is the body rotation rate. q naught initial incidence, that is α naught xz, initial mass 13.5 kg that we had used, and inertia was also given. I think I missed opening this block in the previous lecture. Anyhow, we need to enter the parameters similarly for six DOF as well. So here we have again x e y z e. Body components u, v, and w; roll, pitch, and yaw p, q, r; initial mass and inertia matrix will be here—not a scalar but a matrix form. All right, and we need to give as the input as f x, f y, f z, which is again a matrix. Similarly, m x, m y, and m z. Now, let us look at what is underneath this mask. So here, the equations we are expecting are already integrated in this model. Let me right-click this model and check. Look under the mask. So, control u. All right. I'll just enter control u. So this is what is underneath this mask. I just pressed the space. Button on the keyboard, and we have the full view of the model over here.

(Refer Slide Time: 48:04)

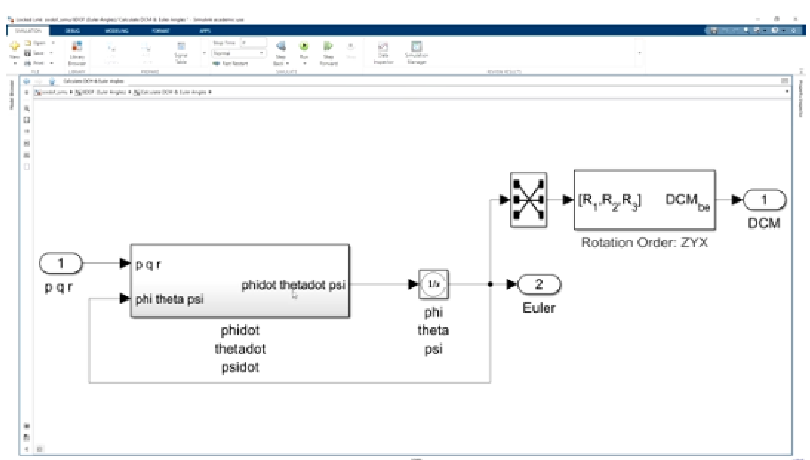


Now, let us first focus on, to get confidence in this model. Let us first focus on $\dot{\theta}$.

$$\dot{\theta} = q\cos\phi - r\sin\phi$$

Let us check do we have this equation in this block or not. Of course, it will be there. We just need to look carefully. So, if we see this block Euler angles, if we are integrating these Euler angles, we should get Euler angles, control U, space button, yes. So, if we see, if go back to the equation, what are the inputs required? Inputs required are the Euler angle itself, ϕ, θ and ψ and the rates, angular rates, P, Q and R. And this equation $\dot{\theta}$ should be integrated. All right. So where we have an integration symbol over here as well. And the inputs are of course the angular rates PQR and the Euler angles here. So when we integrate this we should get ϕ θ and ψ. Let us see do we have that $\dot{\theta}$ equation here or not.

(Refer Slide Time: 49:03)

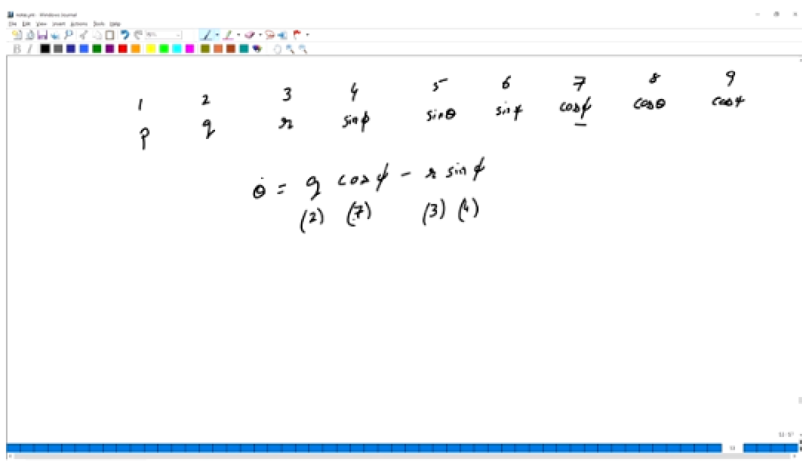


So $\dot{\theta}$ is this equation. All right. The first inputs are P, Q, and R. So let me write over here. So there are actually nine inputs: P, Q, and R; then ϕ, θ, and ψ; sine of ϕ, sine θ, sine ψ; cos of ϕ, cos θ, cos ψ. Let me write here. So we have nine inputs: 1, 2, 3, 4, 5, 6, 7, 8, and 9. The first three inputs are P, Q, and R. Let me write over here: P, Q, R. Next inputs are sine of ϕ, sine of θ, sine of ψ. All right. So, I have sine of ϕ, sine of θ, sine of ψ. All right. Next, we have cos of ϕ, cos θ, cos ψ. cos ϕ, cos θ, cos ψ. All right. Now, let us see the θ dot expression. $\dot{\theta}$, it is saying the inputs are 2 into 7. Θ dot expression is, let me rewrite it here,

$$\dot{\theta} = q\cos\phi - r\sin\phi$$

so it should be the Q variable is 2, input is 2, cos ϕ is here is 7, R is 3, and sin ϕ is 4. So, you should have input of u of 2 into u of 7. So, we have here u of 2 into u of 7 minus u of 3 and u of 4. So, this is how we check the equations.

(Refer Slide Time: 51:03)

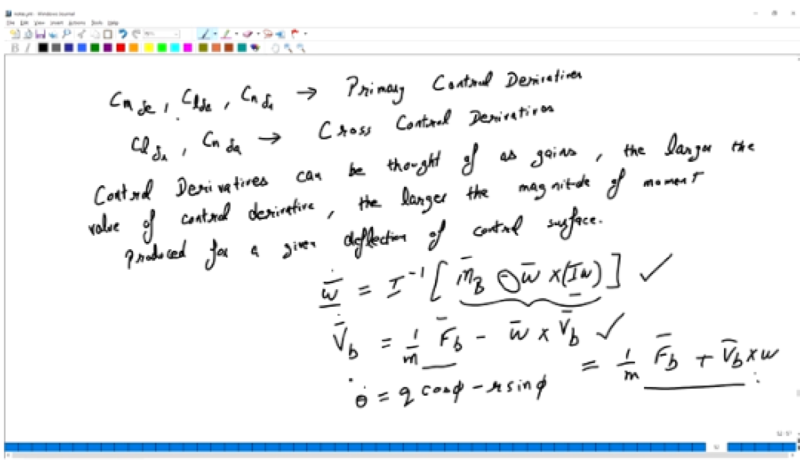


So, similarly, you can check it for ϕ dot and ψ dot. All right. Let us check, give a check on another equation. That I have written over here. All right, so omega is nothing but p dot, q dot, and r dot equals to I inverse mb minus omega cross I omega. So let us see where this block is actually calculating the rates. So here we have p dot, q dot, and r dot. So here it is actually integrating. To get p, q, and r. All right, let us see what is happening in this block. All right, so we have here I inverse equals to mb minus omega cross I omega. So first, let us multiply I into omega. What do we have here? Yeah. So here we have I into omega. It is multiplying here I into omega, which is again the cross product of PQR cross I omega. Here we have cross product of PQR.

Here, A is what? Here, A is PQR cross I omega. This is cross I omega. And this has I. A negative sign over here—this is a negative sign, so we have a negative sign over here. Perfect. Then the moments are added. Then the moments are added. Yes, we have moments added, and we see here another term, which is I dot. Since the mass is actually fixed, inertia is not varying, so this will be zero in this case. Finally, we have I inverse multiplied by this whole equation. Inverse of this—so here, this is I dot. If you double-click, this is the product block. This is multiplied by this I inverse. This is I inverse. So finally, we get $\dot{p}, \dot{q}, \dot{r}$ equations. Now, let us check this as well— $\dot{u}, \dot{v}, \dot{w}$. Yeah, so here we have velocity in the body x-axis. All right. So here, it is actually integrating this. OK. Now, to check the expression, that is F upon M, F upon M. So here we have.

Yes, we have here forces divided by mass. All right. F upon F minus omega cross VB. That is PQR cross the velocities. Here we have this cross product. But the cross product here is we have PQR cross VB. Here it is A cross B. Here A is UV and W. So that is why he has considered plus over here. See, this is C equals to A into B. So, if we can write this as 1 by M FB plus VB cross omega. That is what is represented over here. All right.

(Refer Slide Time: 54:22)



So, finally, we get velocities in body axis. Likewise, I encourage you to explore this model and plot the six DOF simulation by yourself. So more or less we are done with this six DOF simulation and regarding the examination, There will be no MATLAB or Simulink environment which will be provided. The questions will be such that it can be easily solvable using a simple calculator or in pen and paper. If you have any queries or any questions, you are suggested to put it in a discussion forum. In fact, you are encouraged to put any query in the discussion forum. Course instructors and the TAs will ensure to answer your query as quick as possible. And if you would like to reach me, you can contact me at this email ID. P-R-A-B-H-I-I-T-K at the rate gmail.com. So, Prabh IIT Kanpur at gmail.com. So, let us stop it here. Thank you.