The control law looks like (where $\Lambda = (J_v A^{-1} J_v^\top)^{-1}$).

$$\Gamma' = J_v^\top \left(\Lambda \left(k_p(x_d - x) - k_v \dot{x}\right)\right) - g$$

where x_d represents the desired path trajectory and x represents the current positions of the robot. The gravity compensation g is actually handled internally by the robot. Because we only have a need for three degrees of freedom, we made a slight modification to the force in order to reduce the null space movement of the first joint (since we had three degrees of freedom already from our other three joints). When adding our null space damping term, we also exert a joint space force to keep the joint value of the first joint at zero. This means calculating the following for our final control law:

$$\begin{aligned}\Gamma &= \Gamma' + \beta_{\text{ns}}(\Gamma_{\text{ns}} + \Gamma_0) \\ &= \Gamma' + \beta_{\text{ns}}(\Gamma_{\text{ns}} + [k_p q_0 + k_v \dot{q}_0, 0, 0, 0])\end{aligned}$$

Notice that here, we are only implementing joint space control on joint zero and the total sum of the forces makes the robot use the remaining three joints to execute our intended task. Our final parameters for the above task were : $\beta_{\text{ns}} = 3.0, k_p = 200, k_v = 30$. In both the path generation and the ellipse following tasks, the robot moves along a given path and switches direction based on the value of `ALPHA_WAVE_VALUE`. In order to accomplish this, we had to define our x_d equation as follows:

$$\begin{aligned}x_d[0] &= x_{\text{init}}[0] \\ x_d[1] &= x_{\text{init}}[1] + 0.15 \cdot \sin(2\pi f \text{sgn}[\alpha > \alpha_{\text{thresh}}](t - t_{\text{toggle}})) + x_{d,\text{toggle}}[1] \\ x_d[2] &= x_{\text{init}}[2] + 0.15 \cdot \cos(2\pi f \text{sgn}[\alpha > \alpha_{\text{thresh}}](t - t_{\text{toggle}})) + x_{d,\text{toggle}}[2]\end{aligned}$$

where f is the frequency, α is the magnitude of the 10 Hz frequency bin in the EEG signal, $x_{d,\text{toggle}}$ represents the last position the robot was at before toggling, t_{toggle} is either 0 or the last time the robot toggled, and $\text{sgn}[\alpha > \alpha_{\text{thresh}}]$ is +1 when $\alpha > \alpha_{\text{thresh}}$ and -1 otherwise.

For α_{thresh}, we use hysteresis control by defining a max-threshold of $7 \times 10^{-4} \mu$V and a min-threshold of $5 \times 10^{-4} \mu$V when alpha waves are not detected and detected respectively. This allows us to smooth the task over time. Bringing these limits closer together allows us to get finer control but may also allow noise of alpha wave detection to start dominating.

2.6.2 Task 2: Alpha Wave + Accelerometer Control

For our next task, we implement a combination of alpha wave and accelerometer control. We have the WAM move in speeds proportional to the detected alpha wave signal and accelerometer magnitude along the roll axis of the board. To move the robot to either left or right, we tilt our heads to the respective directions. To lift the robot up, we close our eyes and enter a relaxed conscious state. To bring the robot back down, we open our eyes and enter an excited state. We created a very simple demonstration board to show people the robot moving along the different quadrants, which can be seen in the video link below.

Our desired trajectories were very similar to that of Task 1. We define four positions and moved at a speed proportional to the accelerometer magnitude in left/right direction and we used hysteresis control (as we did with Task 1) and used position control to dynamically change $x_d[2]$, all using position control to minimize drift due to the slightly mis-calibrated gravity compensation that would otherwise appear in