HOMEWORK 4

Kinematics, Dynamics and Control of Robots

Table of Contents

1 PD + Inverse Dynamics Control Law

1.1 Control Parameters

The PD + Inverse Dynamics Control Law is given by

$$\boldsymbol{\tau} = C\dot{\boldsymbol{q}} + \boldsymbol{G} + H\left[\ddot{\boldsymbol{q}}_d - K_p(\boldsymbol{q} - \boldsymbol{q}_d) - K_d(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_d)\right] \tag{1.1}$$

The Equation of Motion (EOM) is given by

$$H\ddot{\boldsymbol{q}} + C\dot{\boldsymbol{q}} + \boldsymbol{G} = \boldsymbol{\tau} \tag{1.2}$$

Substituting $\boldsymbol{\tau}$ into the EOM yields

$$H\ddot{\boldsymbol{q}} + C\dot{\boldsymbol{q}} + \boldsymbol{G} = C\dot{\boldsymbol{q}} + \boldsymbol{G} + H\left[\ddot{\boldsymbol{q}}_d - \boldsymbol{K}_p(\boldsymbol{q} - \boldsymbol{q}_d) - K_d(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_d)\right] \tag{1.3}$$

Rearranging and multiplying by H^{-1} from left results in

$$(\ddot{\boldsymbol{q}} - \ddot{\boldsymbol{q}}_d) + K_d(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_d) + K_p(\boldsymbol{q} - \boldsymbol{q}_d) = \boldsymbol{0} \tag{1.4}$$

Defining an error vector with respect to the desired joint position $\boldsymbol{e} \equiv \boldsymbol{q} - \boldsymbol{q}_d$ yields

$$\ddot{\boldsymbol{e}} + K_d\dot{\boldsymbol{e}} + K_p\boldsymbol{e} = 0 \tag{1.5}$$

Thus, the error dynamics should follow the latter equation, and we can pole-place it as we wish.

We chose a 2nd order Bessel prototype response pole by setting $\omega_0 = 4[rad/s]$ and by comparing coefficients received the following:

$$\boxed{\{K_d\}_{i=1,2,3} = 2 \cdot 0.866 \cdot \omega_0 = 6.928} \quad ; \quad \boxed{\{K_p\}_{i=1,2,3} = \omega_0^2 = 16} \tag{1.6}$$

The reason is because to receive a settling time of 1.5% we would need between 4 and 5-time constants, so for a settling time of $t_s = 1.5[sec] = \frac{5}{\zeta\omega_0}$ we would want at least $\frac{5}{1.5 \cdot 0.8660} < 4 = \omega_0$. We chose the Bessel prototype as opposed to ITAE to minimize oscillations, and by wisely limiting ω_0 we resulted with relatively low control parameters yielding a minimal control effort.

1.2 Dynamic Simulation

To properly answer both questions (1) and (2) we chose to simulate dynamics for 3 seconds, and modified the planned path such that after 2 seconds the position stays at point B, and the velocity and acceleration of the tool is zero in all directions. This ensured that when we planned the controller, we could check that it satisfied all criteria.

We simulated the response via ode45[1] given an error in the initial position of the tool of $1cm$ upward ($+\hat{z}_0$) from point A, resulting in the following convergence. It is shown that the tracking error converges to 0.15% in less than 1.5 seconds.

[1] We used specific ode45 options as we encountered numerical issues causing unexpected oscillations. "MaxStep" was chosen as dt=0.001, a "RelTol" of 10^{-4} and an "AbsTol" of 10^{-14} were chosen for optimal results.

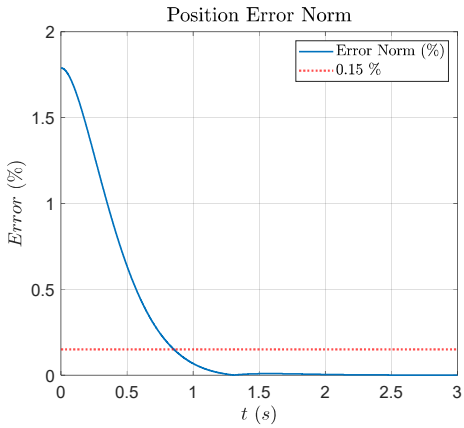


Figure 1.1 - Position Error Norm percent with respect to the total path length.

We approved that the tool position converged to the required position with minimal oscillations by plotting the planned and the simulated task values and their error with respect to the planned values as follows.

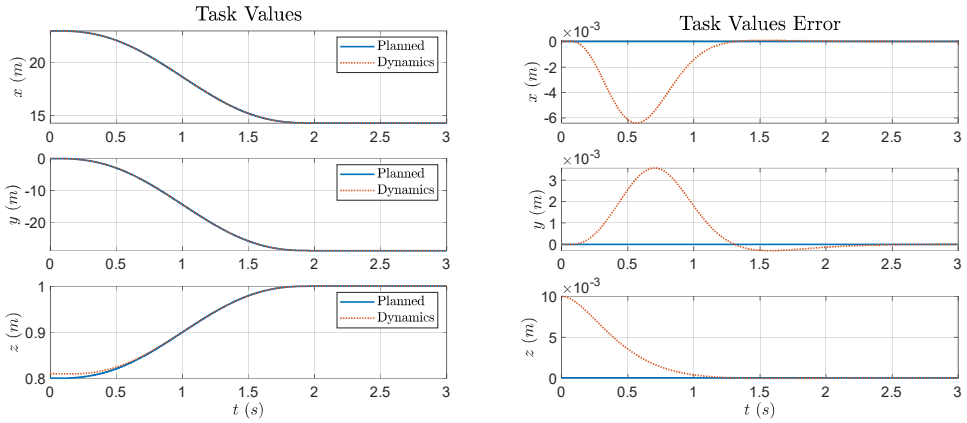


Figure 1.2 - Task values (left) for the planned path and the dynamic simulation. The error of the simulation with respect to the planned task values (right).

We plotted the control torques of the dynamic simulation and the planned torques and found a relatively small difference.

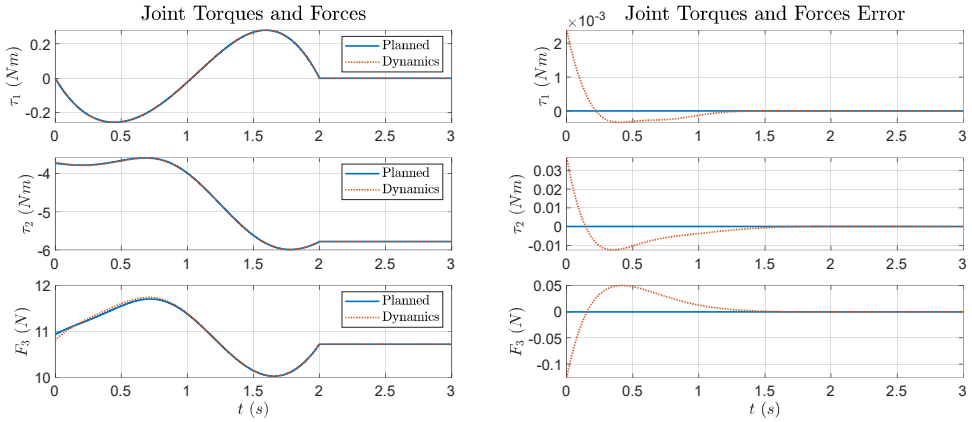


Figure 1.3 - Joint Torques and Forces (left) for the planned path and the dynamic simulation. The error of the simulation with respect to the planned values (right).

The joint values error is given below. The chosen prototype for the error dynamics was successful, because there are no oscillations, and the convergence is fast as expected from a 2nd order Bessel prototype.

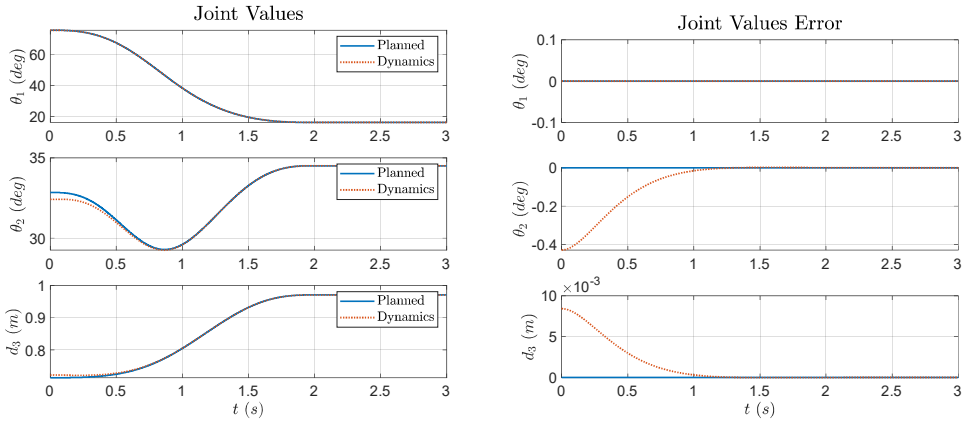


Figure 1.4 - Joint values of the dynamic system and the planned system (left), and the error (right).

1.3 Dynamic Simulation (no load mass)

We repeated the simulation with no load mass and kept the same control law. We made sure that the actual dynamics simulation is with $M = 0kg$ and that the control torques act as if there is $M = 0.5kg$.

Assuming the control law consists of some uncertainty in the free term $\boldsymbol{\Delta G}$ as follows

$$\boldsymbol{\tau} = C\dot{\boldsymbol{q}} + \boldsymbol{G} + \boldsymbol{\Delta G} + H\left[\ddot{\boldsymbol{q}}_d - K_p(\boldsymbol{q} - \boldsymbol{q}_d) - K_d(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_d)\right] \tag{1.7}$$

The resulting error dynamics will be given by

$$\ddot{\boldsymbol{e}} + K_d\dot{\boldsymbol{e}} + K_p\boldsymbol{e} = -H^{-1}\boldsymbol{\Delta G} \tag{1.8}$$

The Error in joint values with respect to the planned motion are plotted below. It is shown that the controller no longer stabilizes the system about the desired values because it assumed that the dynamics are known, yet it has uncertainty in the mass. This causes a steady state error in joint values because the error dynamics will now follow a nonhomogeneous EOM due to this uncertainty.

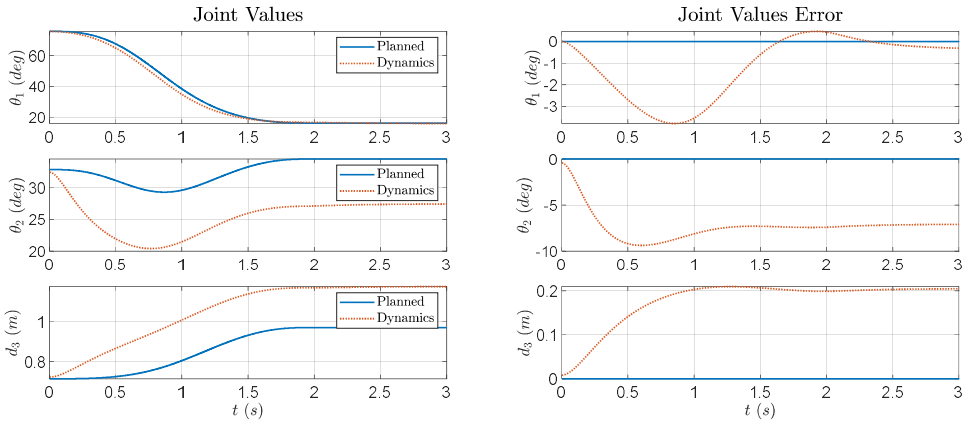


Figure 1.5 - The joint values of the simulation compared with the planned motion (left), and the error (right).

The error norm in tool position shows a large steady state error indicating that the Inverse Dynamics control law lacks robustness.

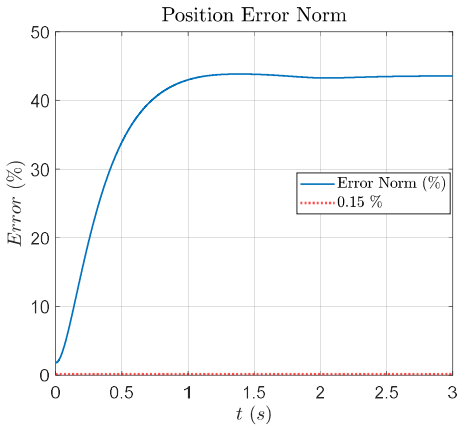


Figure 1.6 - Position error norm as percentage of total path length.

The control torques and forces show different values compared to the previous simulation because the controller will always overcompensate and undercompensate for the errors due to the uncertainty in the dynamics of the system. Also here, there is a steady state error due to this type of uncertainty.

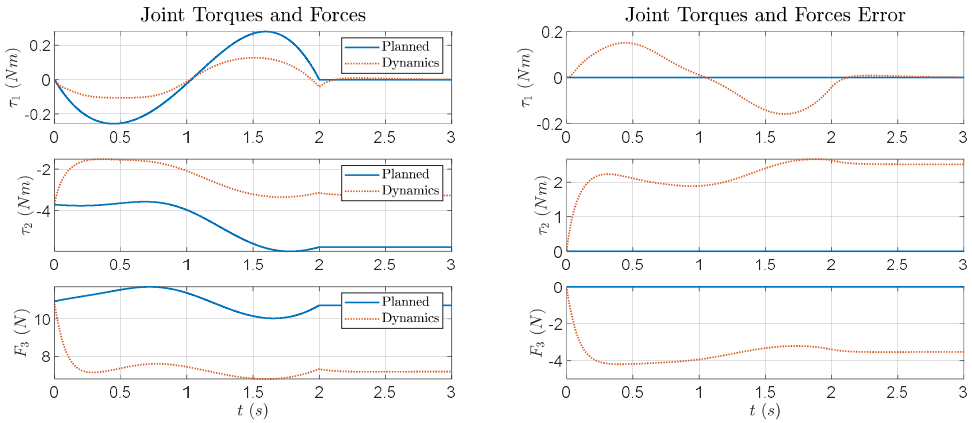


Figure 1.7 - Joint Torques and Forces (left) for the planned path and the dynamic simulation. The error of the simulation with respect to the planned values (right).

2 PD + G Control Law

2.1 Control parameters

The PD + G Control Law is given by

$$\boldsymbol{\tau} = \boldsymbol{G} - K_d(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_d) - K_p(\boldsymbol{q} - \boldsymbol{q}_d) \tag{2.1}$$

The Equation of Motion (EOM) is given by

$$H\ddot{\boldsymbol{q}} + C\dot{\boldsymbol{q}} + \boldsymbol{G} = \tau \tag{2.2}$$

Substituting τ into the EOM yields

$$H\ddot{\boldsymbol{q}} + C\dot{\boldsymbol{q}} + \boldsymbol{G} = \boldsymbol{G} - K_d(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_d) - K_p(\boldsymbol{q} - \boldsymbol{q}_d) \tag{2.3}$$

Rearranging and assuming point stabilization $\boldsymbol{q}_d = const$ and that $\boldsymbol{e} \equiv \boldsymbol{q} - \boldsymbol{q}_d$ yields

$$H\ddot{\boldsymbol{e}} + (K_d + C)\dot{\boldsymbol{e}} + K_p\boldsymbol{e} = 0 \qquad (2.4)$$

Given that K_d and K_p are positive definite and symmetric matrices, the set point (only when arriving at point B) stabilization is locally stable according to the lecture. We chose the control values after some iterations to satisfy conditions as follows.

$$K_p = \begin{bmatrix} 1000 & 0 & 0 \\ 0 & 1200 & 0 \\ 0 & 0 & 1100 \end{bmatrix} > 0 \quad ; \quad K_d = \begin{bmatrix} 30 & 0 & 0 \\ 0 & 40 & 0 \\ 0 & 0 & 50 \end{bmatrix} > 0 \qquad (2.5)$$

We increased $\{K_p\}_i$ values from (1.6) until reaching a satisfactory convergence time. To minimize oscillations in joint values error we increased values for $\{K_d\}_i$ until reaching satisfactory results.

2.2 Dynamic simulation

The results are plotted below. We used a similar approach as previously described by setting the simulation to 3 seconds in advance to satisfy both questions (1) and (2).

The joint values converge to the desired planned values with nearly no oscillations, and after 2 seconds they reach the desired values.

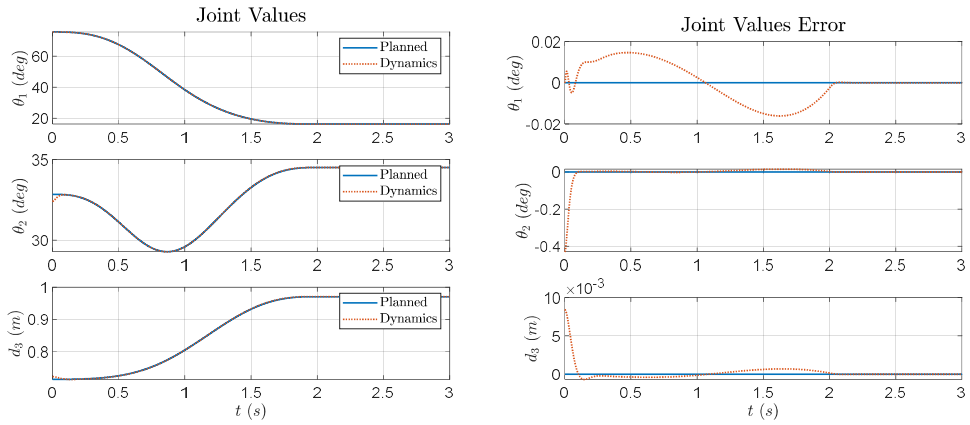


Figure 2.1 - Joint values of the dynamic system and the planned system (left), and the error (right).

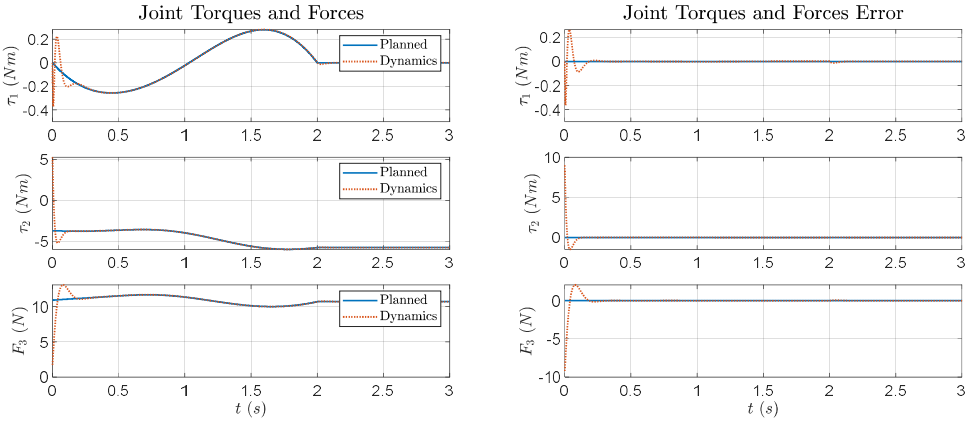


Figure 2.2 - Joint Torques and Forces (left) for the planned path and the dynamic simulation. The error of the simulation with respect to the planned values (right).

We also added the task values plots to show the effectiveness of the control law in the task space.

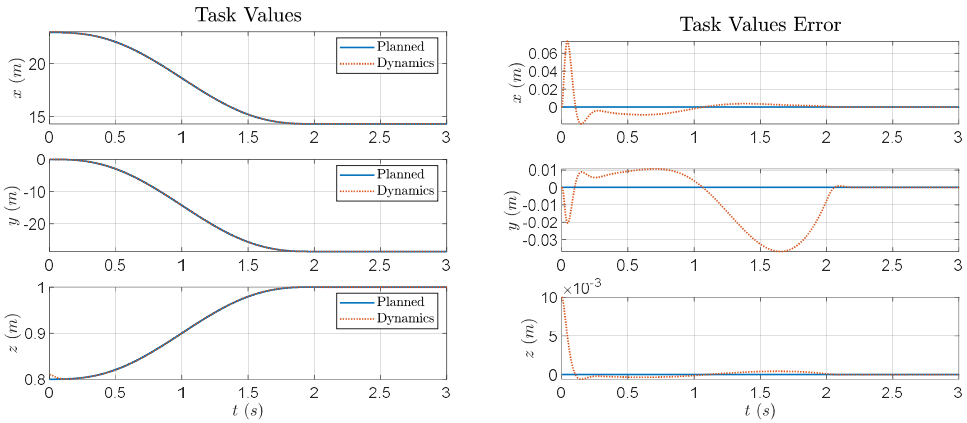


Figure 2.3 - Task values (left) for the planned path and the dynamic simulation. The error of the simulation with respect to the planned task values (right).

It is also shown that the position error norm converges within less than 1.5 seconds as required.

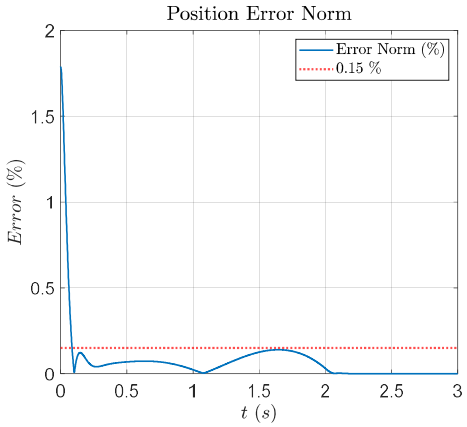


Figure 2.4 - Position Error Norm percent with respect to the total path length.

2.3 Dynamic simulation (no load mass)

Assuming uncertainty in the mass values introduces a steady state error as shown in the position error norm and in the joint values error plots as expected.

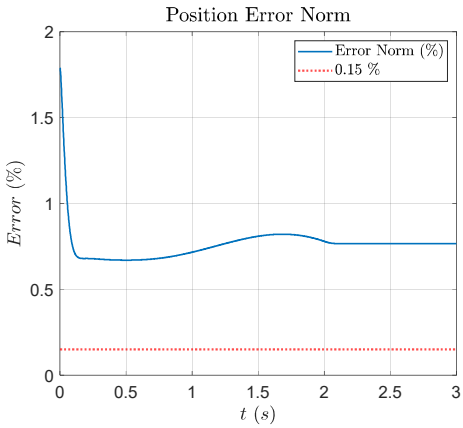


Figure 2.5 - Position Error Norm percent with respect to the total path length.

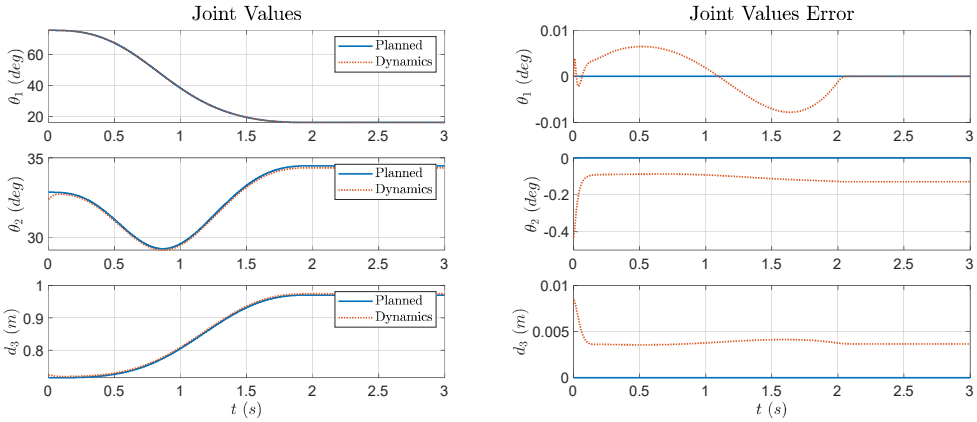


Figure 2.6 - Joint values of the dynamic system and the planned system (left), and the error (right).

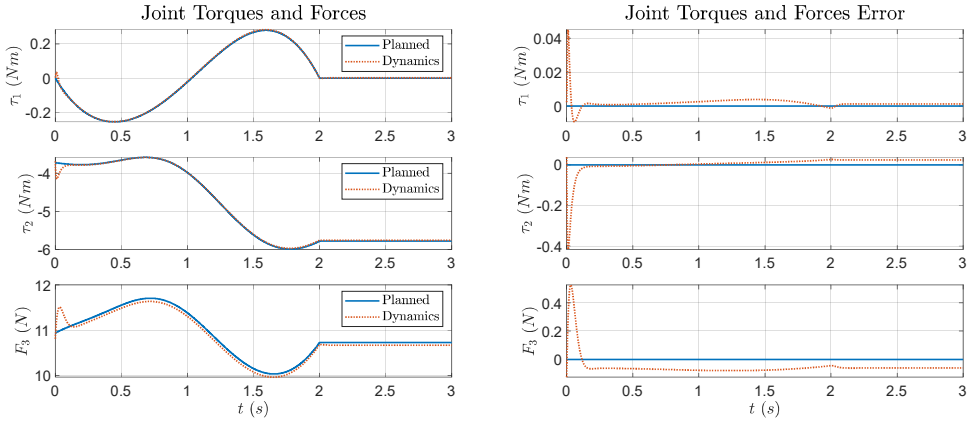


Figure 2.7 - Joint Torques and Forces (left) for the planned path and the dynamic simulation. The error of the simulation with respect to the planned values (right).

3 PID Control Law

3.1 Control parameters

For PID control we modified and augmented the state vector to include the integral over the error.

$$\boldsymbol{X} = \begin{bmatrix} \int e_1 \\ \int e_2 \\ \int e_3 \\ e_1 \\ e_2 \\ e_3 \\ \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} \quad ; \quad \dot{\boldsymbol{X}} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \\ \ddot{e}_1 \\ \ddot{e}_2 \\ \ddot{e}_3 \end{bmatrix} \tag{3.1}$$

We used the control law as follows

$$\boldsymbol{\tau} = -K_p \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} - K_d \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} - K_I \begin{bmatrix} \int_0^t e_1 dt \\ \int_0^t e_2 dt \\ \int_0^t e_3 dt \end{bmatrix} \tag{3.2}$$

After some trial and error, we derived the following parameters.

$$K_p = \begin{bmatrix} 1300 & 0 & 0 \\ 0 & 1700 & 0 \\ 0 & 0 & 1100 \end{bmatrix}; K_d = \begin{bmatrix} 50 & 0 & 0 \\ 0 & 60 & 0 \\ 0 & 0 & 50 \end{bmatrix}; K_I = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 4000 & 0 \\ 0 & 0 & 2000 \end{bmatrix} \tag{3.3}$$

We tuned the parameters K_p and K_d in a similar way mentioned in Control parameters2.1. In addition, we increased K_I to diminish the steady state error.

3.2 Dynamic simulation

We pseudo-randomly chose a mass between 0 and 1 using a seed of 1. ($M = 0.417kg$). The simulated results[2] are given below. The joint values converge to the desired values with relatively low oscillations. The task values reach point B within 2 seconds and stay there asymptotically. In addition, the position error norm is within the required values.

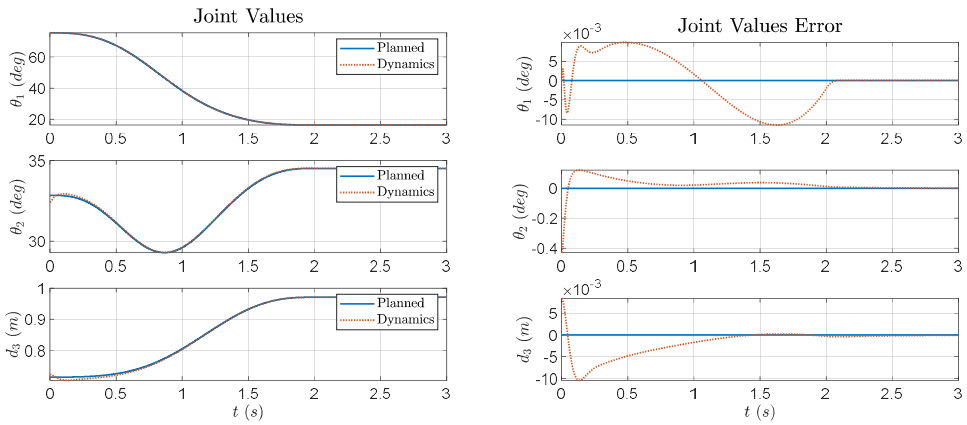


Figure 3.1 - Joint values of the dynamic system and the planned system (left), and the error (right).

[2] Due to the requirements of the function "tou_cont(q, qdot, t)", it is not simple to calculate the control torques without knowing the state vector (specifically, the integral over the error). Since ode45 uses a Runga Kutta method, meaning that it uses different values of "t" in each calculation, one does not simply "calculate" tau based only on (q,qdot,t). To solve this issue, we chose to save the values of "tau" for each timestep and let "tou_cont" function access the correct values post simulation. Some may suggest using "persistent", or "global" within the simulation, and update the next state with the previous state, but this is not correct because it will just end up calculating a different approximation for "tau" because it will not follow the same interpolations made by ode45.

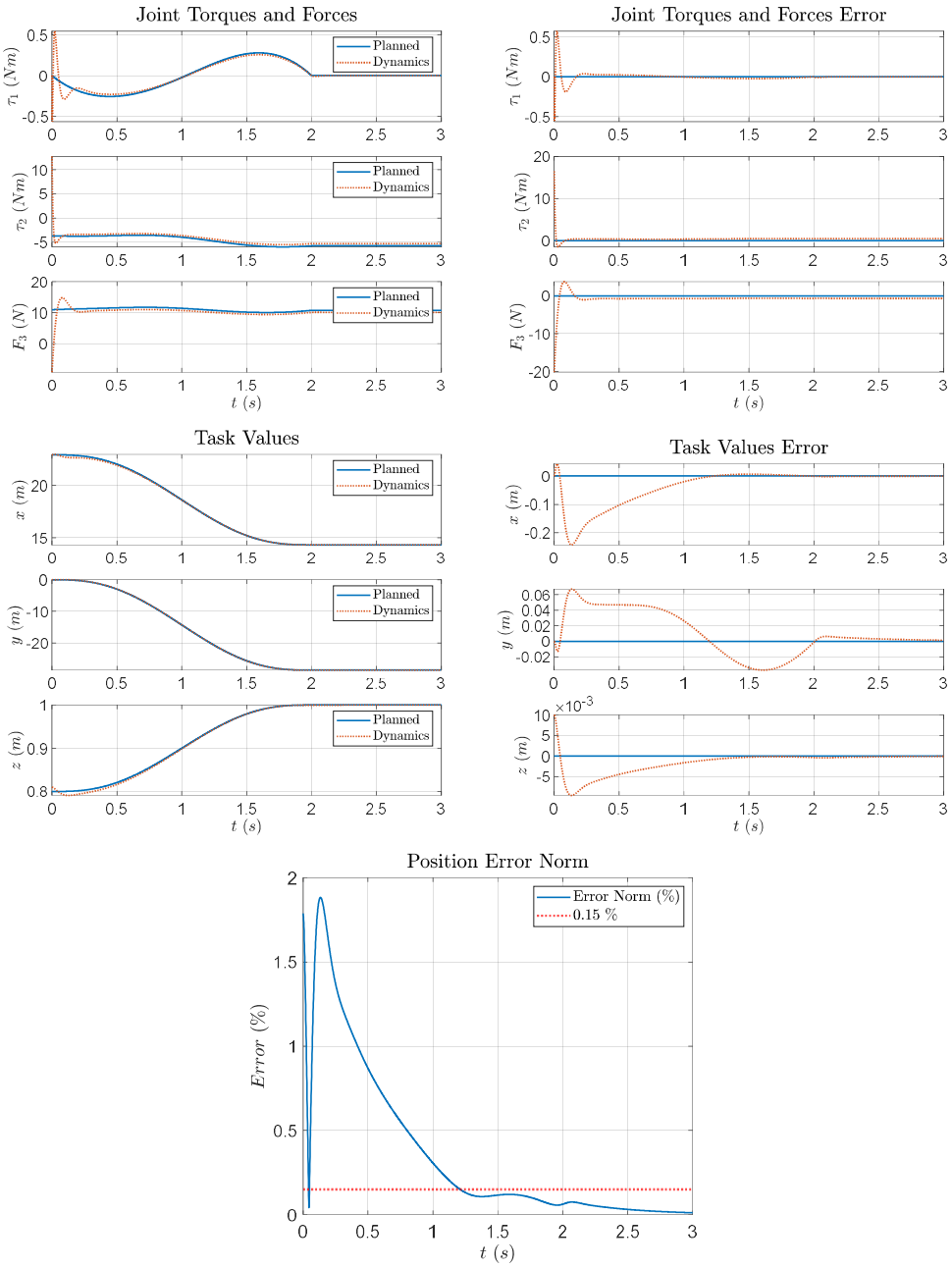


Figure 3.2 – results for the simulation for PID control law.

3.3 Dynamic simulation (no load mass)

The resulting simulation is given below. The joint values converge to the desired values as before, and the position error norm is also within the required values. In addition, it is evident that the steady state joint torques and forces errors are larger than before, and the reason is simply because now the difference between the “planned” mass of 0.5kg to the real mass of 0kg is larger than the difference before (between 0.5kg and 0.417kg). The effect is less significant than other control laws since this law is unaware of the mass to begin with. Therefore, by changing the mass, the PID control law effectively does a similar (yet different) job.

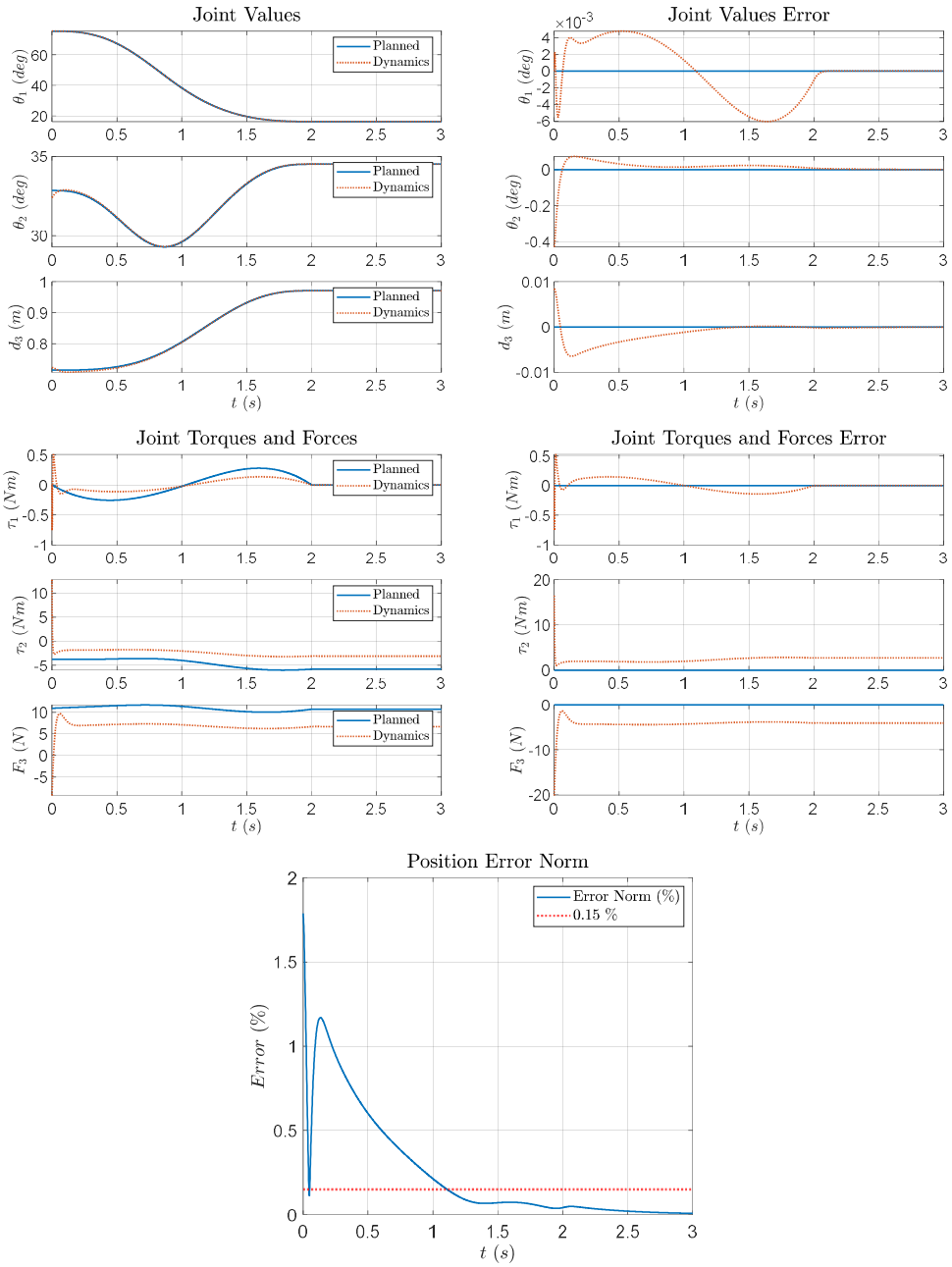


Figure 3.3 - results for the no load simulation for PID control law.

4 MINMAX Control Law

4.1 Control parameters

Given the EOM

$$H\ddot{\boldsymbol{q}} + C\dot{\boldsymbol{q}} + \boldsymbol{G} = \boldsymbol{\tau} \tag{4.1}$$

We define the state vector

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{e} \\ \dot{\boldsymbol{e}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{q} - \boldsymbol{q_d} \\ \dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_{\boldsymbol{d}} \end{bmatrix}_{6x1} \tag{4.2}$$

We then define a weighted error $\boldsymbol{s}$, where K_{3x3} is positive definite.

$$\boldsymbol{s} = \dot{\boldsymbol{e}} + K\boldsymbol{e} \tag{4.3}$$

Then, we define a corrective velocity vector $\dot{\boldsymbol{q}}_r$

$$\dot{\boldsymbol{q}}_r \equiv \dot{\boldsymbol{q}} - \boldsymbol{s} = \dot{\boldsymbol{q}} - (\dot{\boldsymbol{e}} + K\boldsymbol{e}) = \dot{\boldsymbol{q}}_{\boldsymbol{d}} - K\boldsymbol{e} \tag{4.4}$$

$$\therefore \ddot{\boldsymbol{q}}_r = \ddot{\boldsymbol{q}}_{\boldsymbol{d}} - K\dot{\boldsymbol{e}} \tag{4.5}$$

Next, will choose a candidate Lyapunov function $V(\boldsymbol{x}, t)$

$$V(\boldsymbol{X}, t) = \frac{1}{2}\boldsymbol{s}^{\boldsymbol{T}} H\boldsymbol{s} + \frac{\boldsymbol{1}}{\boldsymbol{2}}\boldsymbol{e}^{\boldsymbol{T}} P\boldsymbol{e} \tag{4.6}$$

Then, according to the lecture, for a matrix P that satisfies the following

$$P = P^T > 0 \quad \text{and} \quad KP = P^T K > 0 \tag{4.7}$$

the function V is positive for any nonzero state $\boldsymbol{X}$.

Next, assuming $\boldsymbol{F}_e = \boldsymbol{0}$ we define $\boldsymbol{\eta}$ as

$$\boldsymbol{\eta} = -H\ddot{\boldsymbol{q}}_r - C\dot{\boldsymbol{q}}_r - \boldsymbol{G} + P\boldsymbol{e} = -H(\ddot{\boldsymbol{q}}_d - K\dot{\boldsymbol{e}}) - C\dot{\boldsymbol{q}}_r - \boldsymbol{G} + P\boldsymbol{e} \tag{4.8}$$

We will then split $\boldsymbol{\eta}$ into the known and unknown dynamics. Since we don't know the mass, we will assume it is 0 and find H_0, C_0, G_0 accordingly. Then for the worst case we will use $M = 1kg$.

$$\boldsymbol{\eta} = \tilde{\boldsymbol{\eta}} + \boldsymbol{\eta}_0 = -\left(\tilde{H} + H_0\right)\ddot{\boldsymbol{q}}_r - \left(\tilde{C} + C_0\right)\dot{\boldsymbol{q}}_r - \left(\tilde{\boldsymbol{G}} + \boldsymbol{G_0}\right) + P\boldsymbol{e} \tag{4.9}$$

$$H = \tilde{H} + H_0 \quad ; \quad C = \tilde{C} + C_0 \quad ; \quad \boldsymbol{G} = \tilde{\boldsymbol{G}} + \boldsymbol{G_0} \tag{4.10}$$

Thus,

$$\tilde{\boldsymbol{\eta}} = -\tilde{H}\ddot{\boldsymbol{q}}_r - \tilde{C}\dot{\boldsymbol{q}}_r - \tilde{\boldsymbol{G}} \tag{4.11}$$

$$\boldsymbol{\eta_0} = -H_0\ddot{\boldsymbol{q}}_r - C_0\dot{\boldsymbol{q}}_r - \boldsymbol{G_0} + P\boldsymbol{e} \tag{4.12}$$

The values for $\boldsymbol{\eta_0}$ are known, and the values for $\tilde{\boldsymbol{\eta}}$ are bounded by the case $M = 1kg$. This is evident in all three cases by the fact that the values in (4.13),(4.14) and (4.15) are multiplied by M as shown below.

$$\tilde{H} = \begin{pmatrix} M\left(L^2 + {d_3}^2 \sin(\theta_2)^2\right) & -L\,M\,d_3\cos(\theta_2) & -L\,M\sin(\theta_2) \\ -L\,M\,d_3\cos(\theta_2) & M\,{d_3}^2 & 0 \\ -L\,M\sin(\theta_2) & 0 & M \end{pmatrix} \tag{4.13}$$

$$\tilde{C} = \begin{pmatrix} \frac{M d_3 \left(\dot{d}_3 - \dot{d}_3 \cos(2\theta_2) + d_3 \dot{\theta}_2 \sin(2\theta_2)\right)}{2} & M\left(\dot{\theta}_1 \cos(\theta_2) \sin(\theta_2) d_3^2 + L\dot{\theta}_2 \sin(\theta_2) d_3 - L\dot{d}_3 \cos(\theta_2)\right) & -M\left(d_3 \dot{\theta}_1 \cos(\theta_2)^2 + L\dot{\theta}_2 \cos(\theta_2) - d_3 \dot{\theta}_1\right) \\ -\frac{M d_3^2 \dot{\theta}_1 \sin(2\theta_2)}{2} & M d_3 \dot{d}_3 & M d_3 \dot{\theta}_2 \\ M d_3 \dot{\theta}_1 (\cos(\theta_2)^2 - 1) & -M d_3 \dot{\theta}_2 & 0 \end{pmatrix} \quad (4.14)$$

$$\tilde{\boldsymbol{G}} = \begin{pmatrix} 0 \\ -M d_3 g \sin(\theta_2) \\ M g \cos(\theta_2) \end{pmatrix} \quad (4.15)$$

We will define the bound $\tilde{\rho}$ for $\tilde{\boldsymbol{\eta}}$ by setting a factor of $\beta > 1$ and using the worst case $M = 1kg$.

$$\|\tilde{\boldsymbol{\eta}}\| = \left\|-\tilde{H}\ddot{\boldsymbol{q}}_r - \tilde{C}\dot{\boldsymbol{q}}_r - \tilde{\boldsymbol{G}}\right\| \le \left\|\tilde{H}\ddot{\boldsymbol{q}}_r\right\| + \left\|\tilde{C}\dot{\boldsymbol{q}}_r\right\| + \left\|\tilde{\boldsymbol{G}}\right\| \le \cdots$$
$$\le \left\| \underbrace{\tilde{H}}_{M=1} \ddot{\boldsymbol{q}}_r \right\| + \left\| \underbrace{\tilde{C}}_{M=1} \dot{\boldsymbol{q}}_r \right\| + \left\| \underbrace{\tilde{\boldsymbol{G}}}_{M=1} \right\| = \frac{\tilde{\rho}}{\beta} \quad (4.16)$$

Finally, we will choose ρ_u as

$$\rho_u = \max\left\{0, \frac{1}{\|\boldsymbol{s}\|+\delta}\left(\boldsymbol{s}^T \boldsymbol{\eta_0} - \boldsymbol{e}^{\boldsymbol{T}} K^T P \boldsymbol{e}\right) + \tilde{\rho}\right\} \quad (4.17)$$

The tracking stabilization MINMAX control law (with the regularization δ) is then given by

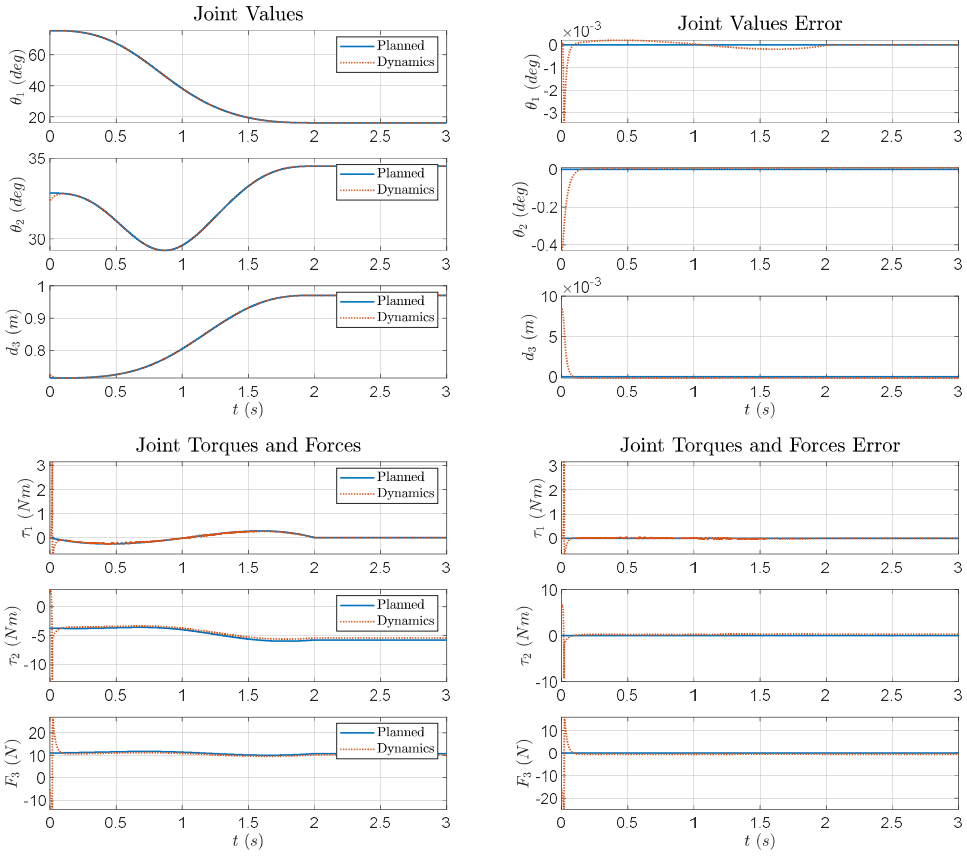
$$\boldsymbol{\tau} = -\rho_u \cdot \frac{\boldsymbol{s}}{\|\boldsymbol{s}\|+\delta} \quad (4.18)$$

The parameters found by trial and error are given below. We found that larger values of K, P values yielded faster responses, and that decreasing δ caused potentially larger control torques resulting in faster responses.

$$K = \begin{bmatrix} 50 & 0 & 0 \\ 0 & 30 & 0 \\ 0 & 0 & 50 \end{bmatrix}; P = \begin{bmatrix} 30 & 0 & 0 \\ 0 & 30 & 0 \\ 0 & 0 & 30 \end{bmatrix}; \delta = 0.005; \beta = 1.4 \quad (4.19)$$

4.2 Dynamic simulation

We pseudo-randomly chose a mass between 0 and 1 using a seed of 2. ($M = 0.436kg$). The simulated results are given below. The joint values converge to the desired values with relatively low oscillations. The task values reach point B within 2 seconds and stay there asymptotically. In addition, the position error norm is within the required values.



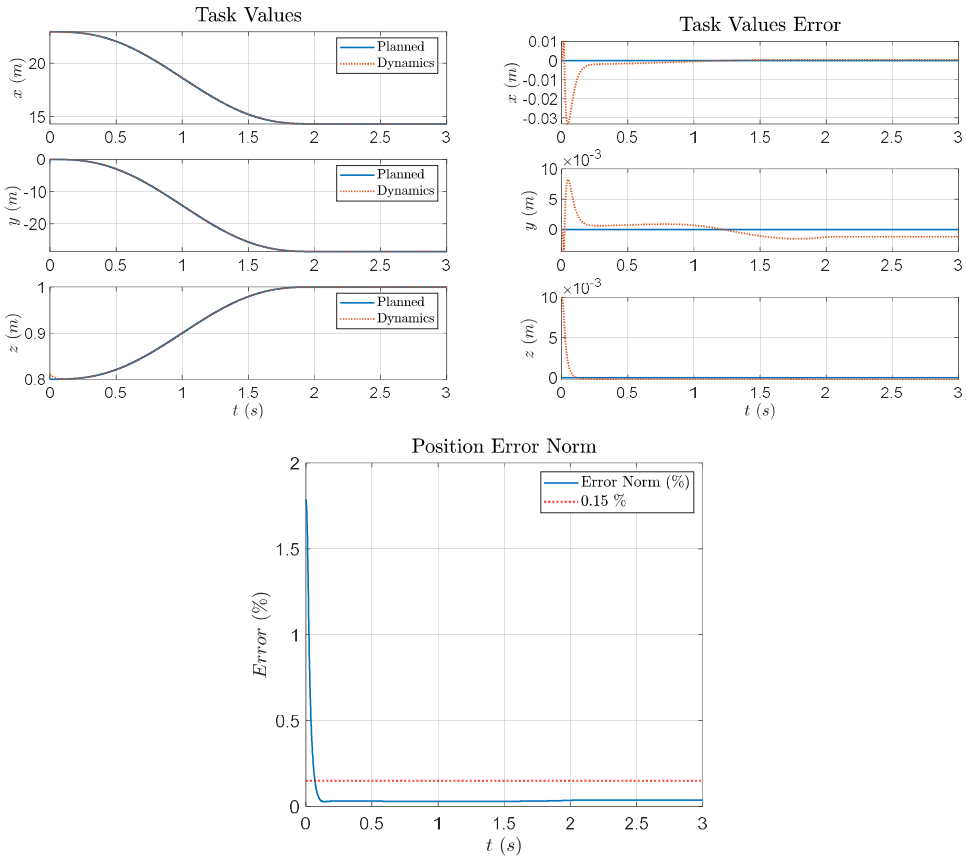


Figure 4.1 - results for the simulation for MINMAX control law.

4.3 Dynamic simulation (no load mass)

The no load simulation results are given below. The joint values converge to the desired values, and the position error norm reaches below the required values very fast (0.1 seconds). In addition, the joint torques are in the same order of magnitude as before. The effect of "no load" vs "load" is not very significant because the MINMAX control law takes the worst-case uncertainty into account by design. Therefore, the convergence in both cases is in a similar fashion.

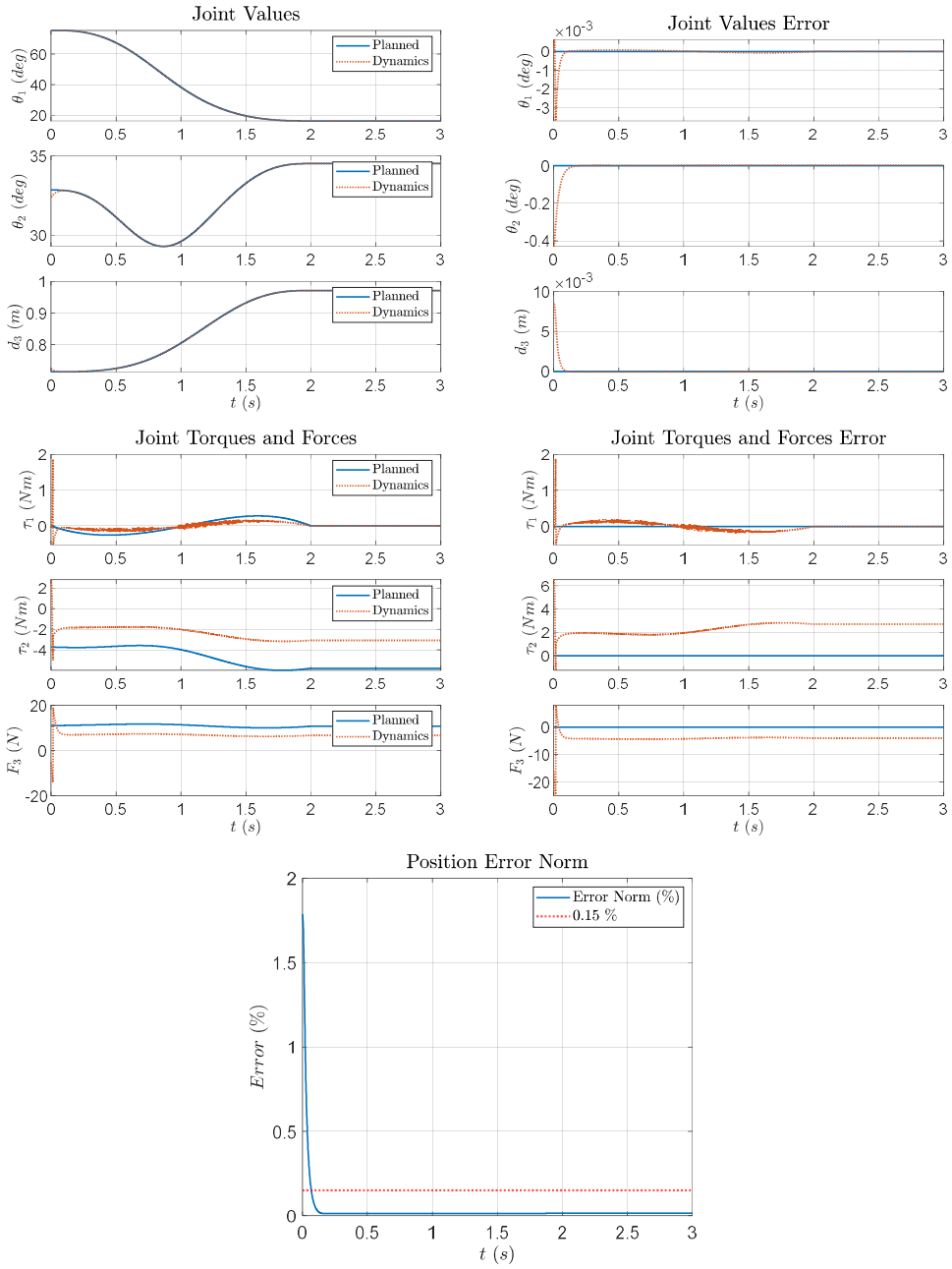


Figure 4.2 - results for the no load simulation for the MINMAX control law.

5 Conclusion

5.1 Complexity

The least complex law is PID. This is because it assumes no knowledge whatsoever of the dynamics of the system and has very few calculation steps. The most complex control law is the MINMAX law, as it is required to make multiple calculations in each step based on the known and bounded dynamics matrices.

5.2 IC error sensitivity

Considering IC error sensitivity only, assuming perfect knowledge of the model, the most powerful control law is the PD + Inverse Dynamics law. This is because we can choose the error dynamics as we wish, and thus reach the smoothest responses with the least amount of control input (recall the Bessel prototype choice). This is shown by the fact that for all four control laws, the simulation that took the least amount of control effort to reach the desired values was the first law (see section 1.2). However, note that when combining this error with model uncertainties, this control law resulted in an utter failure with up to a 40% error (see 1.3).

5.3 Control Effort

The control parameters are smallest for the PD + Inverse Dynamics law and are the largest for the PID law. This is because in the PID law, no model knowledge is considered as opposed to PD. In addition, unlike MINMAX, there is no uncertainty bound to which we can compare to lower the control parameters.

When looking at the control torques and forces, we see that the largest values are for a combination of MINMAX and PID. The largest torque is that of the PID law, and the largest force is that of the MINMAX controller. (see $\tau_2 \approx 10Nm$ in Figure 3.3 and $F_3 \approx 20N$ in Figure 4.2)

5.4 Robustness

The PD + Inverse Dynamics control law is the least robust, as it assumes it knows all the model parameters. The PD + G is slightly less susceptible to uncertainties, but when the parameters are within the vector G it is. The PID control law is more robust than the former two laws, however it can lose stability, so it is essential to test before deploying the law. The MINMAX control law is the most robust of them all since it is designed to oppose errors while considering model parameter uncertainty.