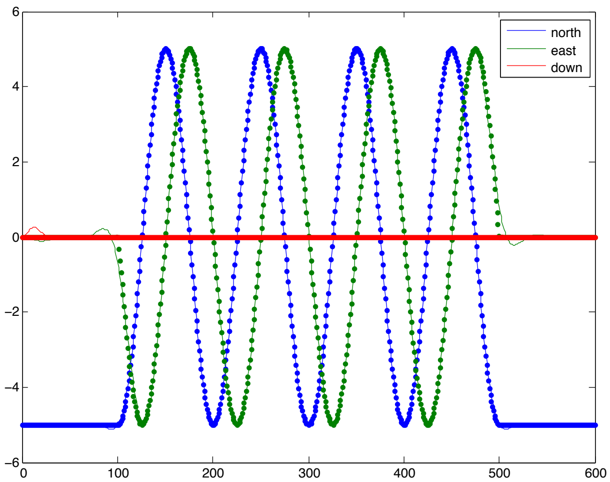
16-899C: Homework #4

William Martin and Daniel Silva

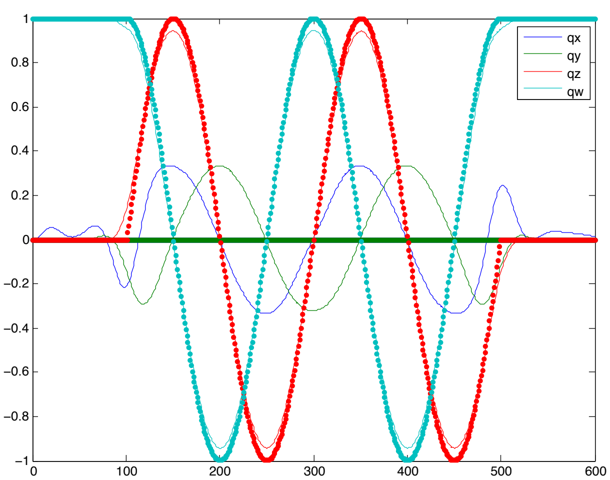
Problem 2: iLQR

(a)

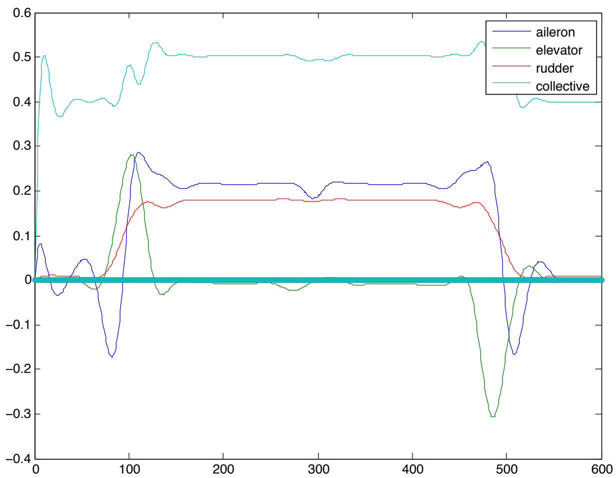
N E D:



quaternion:



u_prev:



(b) Our N, E, D trajectories are followed extremely closely, deviating only 20cm at the beginning and nearly 0cm during the trajectory. Our quaternion matches closely for qz and qw, but there is a deviation of ± 0.3 for qx and qy. qx and qy roughly correspond to the pitch and roll of the helicopter. This error is expected since the helicopter dynamics require the body to roll sideways towards the direction it is moving and to pitch forward to counteract the centrifugal acceleration (induced by the circular motion). u_prev values are all quite low, but are not zero since some control input is required to generate the desired trajectory and reject disturbances.

(c) This DDP code differs from generic DDP code slightly, since it utilizes LQR to generate its control sequences. Normally, we would move backwards in time and use value iteration on a full quadratic model to define our control sequence. Instead, this code assumes a simple linear model by linearizing the proposed trajectory, quadraticizing the cost function, and then running LQR on the linearized system about this trajectory.

The magic factor dictates the weight of the desired state of our system versus the actual simulated state of our system. This allows us follow the desired states more closely during earlier iterations of iLQR when the control would cause the system to fly very far off course. As the loop iterates, the magic factor decreases, which causes the true system dynamics to dominate the results. The magic factor effect is necessary because without it, the linearized trajectory used in LQR would deviate extremely after the first few iterations due to poor initial controls.

One alternative to the magic factor would be to start the iLQR loop with a known controller that does not produce a poor trajectory. This might not be practical, so a better option would be to define a trusted region of the state that limits how far the state can deviate from the desired trajectory before being saturated. This way, we can ensure that the helicopter stays within some region that the linear dynamics will be valid.

Problem 3: Inverse Optimal Control

(a) **Constraint**

$$\sum_{t=1}^{T} x_t^{d\intercal} Q x_t^d + u_t^{d\intercal} R u_t^d \leq \min_{K_{ss}} \sum_{t=1}^{T} x_t{}^\intercal Q x_t + u_t{}^\intercal R u_t$$

where x^d is the demonstrated behavior, x is the optimal control, and K_{ss} is the LQR controller. Q is positive symmetric definite and R is positive symmetric semidefinite.

(b) **Sub-Gradient Update Rule**

To ensure that R is definite we define $\tilde{R} = R + \mathrm{I}$
Sub-gradient:

$$\begin{aligned}\frac{\partial C}{\partial Q} &= \frac{\partial}{\partial Q}\left(\sum_{t=1}^{T} x_t^{d\intercal} Q x_t^d + u_t^{d\intercal} \tilde{R} u_t^d - \sum_{t=1}^{T} x_t^{*\intercal} Q x_t^* + u_t^{*\intercal} \tilde{R} u_t^*\right) \\ &= \left(\sum_{\xi_{demo}} x x^\intercal - \sum_{\xi^*} x x^\intercal\right)\end{aligned}$$

We define our step size as γ, which makes our update rule:

$$Q = Q - \gamma\left(\sum_{\xi_{demo}} x x^\intercal - \sum_{\xi^*} x x^\intercal\right)$$

(c) **Non-Negativity Projection**

Let $\mathcal{C}$ be the cone of positive definite matrices. The projection of a matrix M, denoted by M^+, is defined as:

$$M^+ = \arg\min_{X \in \mathcal{C}} ||M - X||_F$$

Let Q be a symmetric but indefinite matrix.
We know that any real symmetric matrix $Q \in \mathbb{R}^{n\times n}$ can be decomposed (eigendecomposition) as

$$Q = T \Lambda T^T$$

where T is the matrix of the eigenvectors of Q and $\Lambda = diag\{\lambda_1, \ldots, \lambda_n\}$ is a diagonal matrix with the eigenvalues of Q on its main diagonal.
The projection Q^+ on $\mathcal{C}$ can be computed explicitly as follows:

$$Q^+ = T \Lambda^+ T^T$$

where $\Lambda^+ = diag\{\lambda_1^+, \ldots, \lambda_n^+\}$ and $\lambda_i^+ = max\{\lambda_i, 0\}$.
I.e. the projection is by performing an eigen-decomposition and setting all negative eigenvalues to zero on the diagonal matrix Λ.
Let's define $Q^- = Q - Q^+$. By construction, $Q = Q^+ + Q^-$ and $Q^- \notin \mathcal{C}$ (with $(Q^-)^+ = 0$, i.e. Q^- has a zero projection on $\mathcal{C}$). That is true because $Q^+ = (Q + (Q - Q^+))^+ = Q^+ + (Q - Q^+)^+$. Thus, $(Q^-)^+ = (Q - Q^+)^+ = 0$. It is also easy to see that this decomposition of Q into $Q^+ + Q^-$ is unique.
Let $Proj_{\mathcal{C}}(M)$ be the projection of M on $\mathcal{C}$. Let's observe that

$$
\begin{aligned}
\arg\min_{X \in \mathcal{C}} ||M - X||_F &= \arg\min_{X \in \mathcal{C}} ||Proj_{\mathcal{C}}(M) + (M - Proj_{\mathcal{C}}(M)) - X||_F \\
&= \arg\min_{X \in \mathcal{C}} ||(Proj_{\mathcal{C}}(M) - X) + (M - Proj_{\mathcal{C}}(M))||_F \\
&= \arg\min_{X \in \mathcal{C}} ||(Proj_{\mathcal{C}}(M) - X) + (M - Proj_{\mathcal{C}}(M))||_F^2 \\
&= \arg\min_{X \in \mathcal{C}} ||(Proj_{\mathcal{C}}(M) - X)||_F^2 + < (Proj_{\mathcal{C}}(M) - X), (M - Proj_{\mathcal{C}}(M)) >_F \\
&+ ||(M - Proj_{\mathcal{C}}(M))||_F^2 \\
&= \arg\min_{X \in \mathcal{C}} ||(Proj_{\mathcal{C}}(M) - X)||_F^2 + ||(M - Proj_{\mathcal{C}}(M))||_F^2 \\
&= \arg\min_{X \in \mathcal{C}} ||(Proj_{\mathcal{C}}(M) - X)||_F^2 \geq 0
\end{aligned}
$$

Observe that the inner product $< (Proj_{\mathcal{C}}(M) - X), (M - Proj_{\mathcal{C}}(M)) >_F$ is zero (that is why it gets eliminated from step 3 to step 4). The reason is that $(Proj_{\mathcal{C}}(M) - X) \in \mathcal{C}$ and $(M - Proj_{\mathcal{C}}(M)) \notin \mathcal{C}$, with $Proj_{\mathcal{C}}((M - Proj_{\mathcal{C}}(M))) = Proj_{\mathcal{C}}(M) - Proj_{\mathcal{C}}(Proj_{\mathcal{C}}(M))) = Proj_{\mathcal{C}}(M) - Proj_{\mathcal{C}}(M) = 0$. Also, observe that the term $||(M - Proj_{\mathcal{C}}(M))||_F^2$ does not depend on X (that is why it is eliminated from step 4 to step 5). Thus solution for the problem above is $X = Proj_{\mathcal{C}}(M)$, since for that value $||(Proj_{\mathcal{C}}(M) - X)||_F^2 = 0$.
Now, let's observe that by the way Q^+ is constructed, it has only nonnegative eigenvalues. Thus, Q^+ is positive semidefinite and $Q^+ \in \mathcal{C}$.
Now observe that $Q^- = Q - Q^+ = T\Lambda T^T - T\Lambda^+ T^T = T(\Lambda - \Lambda^+)T^T = T\Lambda^- T^T$ where $\Lambda^- = diag\{\lambda_1^-, \ldots, \lambda_n^-\}$ and $\lambda_i^- = min\{\lambda_i, 0\}$. Thus, by construction Q^- has all its eigenvalues nonpositive and thus $Q^- \notin \mathcal{C}$. Besides, observe that $(Q^-)^+ = T(\Lambda^-)^+ T^T = T0T^T = 0$. I.e., the projection of Q^- on $\mathcal{C}$ has norm zero. This shows that Q^+ is actually the projection of Q on $\mathcal{C}$ and thus that $Q^+ = \arg\min_{X \in \mathcal{C}} ||Q - X||_F$.

(d) MATLAB file: *ACLRHW4_2B_Practical.m* After running the script, the derived cost function matrix will be in the MATLAB variable Q and the derived controller will be in the MATLAB variable lqr_traj.K.

Using the controller, target hover state, and starting conditions in *K_ss.mat* we derived a cost function using inverse optimal control. We used a step size of $\gamma = 0.01$ and a LQR cost function of $x^\intercal Q x + u^2$. After approximately 10 iterations, the Q matrix converged. We ensure that the matrix is positive-definite by setting any non-positive eigenvalues to a small positive value.

The derived cost matrix is symmetric and positive-definite, but has many non-zero terms not on the diagonal. This means that some of the cost matrix values influence the product of two state errors. In HW2a, our cost function was of the same form $x^\intercal Qx + u^2$ since our R matrix was 1, but there were only nonzero elements on the Q diagonal, which was

```
[0 0 0 0 1000 1000 1000 1000 1 1 1 1 1 1 1 1 1 1 1 1 0]
```

The IOC cost function puts relatively higher cost on using the second/third inputs and changing the second input. However, the punishment for changing the other inputs is reduced compared to HW2a. The IOC cost function puts a high cost on certain angular deviations and high angular velocities. There is also a high cost on deviation in the northing coordinate and velocity. This is reasonable since the starting state was 10000 units north from the hover position.

The non-negativity fixing is necessary in practice to avoid obtaining a negative cost matrix. Since LQR tries to minimize the cost function, performing it with a negative cost matrix will give poor results since it can grow infinitely negative. If you make the assumption that the demonstrated cost function being learned with IOC is guaranteed positive-definite, then you may be able to avoid the non-negativity fixing, but it is probably still a good idea just in case.