AE 4803 Robotics and Autonomy
Professor Evangelos Theodorou
Homework 1

Luis Pimentel — lpimentel3@gatech.edu
Jackson Crandell — jackcrandell@gatech.edu

October 2, 2020

Part 1.

1.1) First we linearize the dynamics $\frac{d\boldsymbol{x}}{dt} = f(\boldsymbol{x}, \boldsymbol{u}, t)$:

$$
\begin{aligned}
&\frac{d\boldsymbol{x}}{dt} = f(\boldsymbol{x}, \boldsymbol{u}, t) \\
&= f(\boldsymbol{x} + (\bar{\boldsymbol{x}} - \bar{\boldsymbol{x}}), \boldsymbol{u} + (\bar{\boldsymbol{u}} - \bar{\boldsymbol{u}}), t) \\
&= f(\bar{\boldsymbol{x}} + (\boldsymbol{x} - \bar{\boldsymbol{x}}), \bar{\boldsymbol{u}} + (\boldsymbol{u} - \bar{\boldsymbol{u}}), t) \\
&= f(\bar{\boldsymbol{x}} + \delta\boldsymbol{x}, \bar{\boldsymbol{u}} + \delta\boldsymbol{u}, t) \\
&= f(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, t) + \nabla_{\boldsymbol{x}} f \delta\boldsymbol{x} + \nabla_{\boldsymbol{u}} f \delta\boldsymbol{u} \\
&\frac{d\boldsymbol{x}}{dt} - f(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}, t) = \nabla_{\boldsymbol{x}} f \delta\boldsymbol{x} + \nabla_{\boldsymbol{u}} f \delta\boldsymbol{u} \\
&\frac{d\delta\boldsymbol{x}}{dt} = \nabla_{\boldsymbol{x}} f \delta\boldsymbol{x} + \nabla_{\boldsymbol{u}} f \delta\boldsymbol{u}
\end{aligned}
$$

Then we time discretize the linearized dynamics:

$$
\begin{aligned}
&\frac{\delta\boldsymbol{x}(t_{k+1}) - \delta\boldsymbol{x}(t_k)}{t_{k+1} - t_k} = \nabla_{\boldsymbol{x}} f \delta\boldsymbol{x}(t_k) + \nabla_{\boldsymbol{u}} f \delta\boldsymbol{u}(t_k) \\
&\frac{\delta\boldsymbol{x}(t_{k+1}) - \delta\boldsymbol{x}(t_k)}{dt} = \nabla_{\boldsymbol{x}} f \delta\boldsymbol{x}(t_k) + \nabla_{\boldsymbol{u}} f \delta\boldsymbol{u}(t_k) \\
&\delta\boldsymbol{x}(t_{k+1}) - \delta\boldsymbol{x}(t_k) = \nabla_{\boldsymbol{x}} f dt \delta\boldsymbol{x}(t_k) + \nabla_{\boldsymbol{u}} f dt \delta\boldsymbol{u}(t_k) \\
&\delta\boldsymbol{x}(t_{k+1}) = \nabla_{\boldsymbol{x}} f dt \delta\boldsymbol{x}(t_k) + \delta\boldsymbol{x}(t_k) + \nabla_{\boldsymbol{u}} f dt \delta\boldsymbol{u}(t_k) \\
&\delta\boldsymbol{x}(t_{k+1}) = \left(\boldsymbol{I}_{n\times n} + \nabla_{\boldsymbol{x}} f dt\right) \delta\boldsymbol{x}(t_k) + (\nabla_{\boldsymbol{u}} f dt) \delta\boldsymbol{u}(t_k) \\
&\delta\boldsymbol{x}(t_{k+1}) = \boldsymbol{\Phi}(t_k) \delta\boldsymbol{x}(t_k) + \boldsymbol{B}(t_k) \delta\boldsymbol{u}(t_k)
\end{aligned}
$$

Where:

$$\boldsymbol{\Phi}(t_k) = \boldsymbol{I}_{n\times n} + \nabla_{\boldsymbol{x}} f(t_{k+1} + t_k)$$
$$\boldsymbol{B}(t_k) = \nabla_{\boldsymbol{u}} f(t_{k+1} + t_k)$$

1.2) Next we want to express the second-order expansion of $Q(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k))$. First we consider the general 2nd Order Taylor Series Expansion of a single-variable function and a multi-variable function:

$$f(x) \approx f(a) + f'(a)(x-a)^2 + \frac{f''(a)}{2}(x-a)^2 atx = a$$
$$f(x,y) \approx f(a,b) + f_x(a,b)(x-a) + f_y(a,b)(y-b)$$
$$+ \frac{1}{2} f_{xx}(a,b)(x-a)^2 + f_{xy}(a,b)(x-a)(y-b) + \frac{1}{2} f_{yy}(a,b)(y-b)^2$$

We can generalize these forms in matrix-algebra for our state-action value function.

$$Q(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k)) = \mathcal{L}(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k), t_k) + V(\boldsymbol{x}(t_{k+1}), t_{k+1})$$

We will start by writing the expansion of the running cost and the value function individually.

First we do the second order expansion of the runnning cost along the nominal trajectory $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{u}}$:

$$\mathcal{L}(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k), t_k)$$
$$= \mathcal{L}(\boldsymbol{x}(t_k) + (\bar{\boldsymbol{x}}(t_k) - \bar{\boldsymbol{x}}(t_k)), \boldsymbol{u}(t_k) + (\bar{\boldsymbol{u}}(t_k) - \bar{\boldsymbol{u}}(t_k)), t_k)$$
$$= \mathcal{L}(\bar{\boldsymbol{x}}(t_k) + (\boldsymbol{x}(t_k) - \bar{\boldsymbol{x}}(t_k)), \bar{\boldsymbol{u}}(t_k) + (\boldsymbol{u}(t_k) - \bar{\boldsymbol{u}}(t_k)), t_k)$$
$$= \mathcal{L}(\bar{\boldsymbol{x}}(t_k) + \delta\boldsymbol{x}(t_k), \bar{\boldsymbol{u}}(t_k) + \delta\boldsymbol{u}(t_k), t_k)$$
$$= \underbrace{\ell(\bar{\boldsymbol{x}}(t_k), \bar{\boldsymbol{u}}(t_k), t_k)dt}_{\mathcal{L}} + \underbrace{\left(\nabla_{\boldsymbol{x}} \ell(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k), t_k)dt\right)^T}_{\mathcal{L}_{\boldsymbol{x}}} \delta\boldsymbol{x}(t_k) + \underbrace{\left(\nabla_{\boldsymbol{u}} \ell(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k), t_k)dt\right)^T}_{\mathcal{L}_{\boldsymbol{u}}} \delta\boldsymbol{u}(t_k)$$
$$+ \frac{1}{2}\delta\boldsymbol{x}(t_k)^T \underbrace{\left(\nabla_{\boldsymbol{xx}} \ell(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k), t_k)dt\right)}_{\mathcal{L}_{\boldsymbol{xx}}} \delta\boldsymbol{x}(t_k) + \frac{1}{2}\delta\boldsymbol{u}(t_k)^T \underbrace{\left(\nabla_{\boldsymbol{uu}} \ell(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k), t_k)dt\right)}_{\mathcal{L}_{\boldsymbol{uu}}} \delta\boldsymbol{u}(t_k)$$
$$+ \frac{1}{2}\delta\boldsymbol{u}(t_k)^T \underbrace{\left(\nabla_{\boldsymbol{xu}} \ell(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k), t_k)dt\right)}_{\mathcal{L}_{\boldsymbol{xu}}} \delta\boldsymbol{x}(t_k) + \frac{1}{2}\delta\boldsymbol{x}(t_k)^T \underbrace{\left(\nabla_{\boldsymbol{ux}} \ell(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k), t_k)dt\right)}_{\mathcal{L}_{\boldsymbol{ux}}} \delta\boldsymbol{u}(t_k)$$

We denote:

$$\mathcal{L} = \ell(\bar{\boldsymbol{x}}(t_k), \bar{\boldsymbol{u}}(t_k), t_k)dt$$
$$\mathcal{L}_{\boldsymbol{x}} = \nabla_{\boldsymbol{x}}\ell(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k), t_k)dt$$
$$\mathcal{L}_{\boldsymbol{u}} = \nabla_{\boldsymbol{u}}\ell(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k), t_k)dt$$
$$\mathcal{L}_{\boldsymbol{xx}} = \nabla_{\boldsymbol{xx}}\ell(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k), t_k)dt$$
$$\mathcal{L}_{\boldsymbol{uu}} = \nabla_{\boldsymbol{uu}}\ell(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k), t_k)dt$$
$$\mathcal{L}_{\boldsymbol{xu}} = \nabla_{\boldsymbol{xu}}\ell(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k), t_k)dt$$
$$\mathcal{L}_{\boldsymbol{ux}} = \nabla_{\boldsymbol{ux}}\ell(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k), t_k)dt$$

With this we write second order expansion of the running cost as:

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k), t_k)\\
&= \mathcal{L} + \mathcal{L}_{\boldsymbol{x}}^T\delta\boldsymbol{x}(t_k) + \mathcal{L}_{\boldsymbol{u}}^T\delta\boldsymbol{u}(t_k)\\
&+ \frac{1}{2}\delta\boldsymbol{x}(t_k)^T\mathcal{L}_{\boldsymbol{xx}}\delta\boldsymbol{x}(t_k) + \frac{1}{2}\delta\boldsymbol{u}(t_k)^T\mathcal{L}_{\boldsymbol{uu}}\delta\boldsymbol{u}(t_k)\\
&+ \frac{1}{2}\delta\boldsymbol{u}(t_k)^T\mathcal{L}_{\boldsymbol{xu}}\delta\boldsymbol{x}(t_k) + \frac{1}{2}\delta\boldsymbol{x}(t_k)^T\mathcal{L}_{\boldsymbol{ux}}\delta\boldsymbol{u}(t_k)
\end{aligned}$$

Next we do the second order expansion of the value function along the nominal trajectory $\bar{\boldsymbol{x}}$:

$$\begin{aligned}
&V(\boldsymbol{x}(t_{k+1}), t_{k+1})\\
&= V\left(\boldsymbol{x}(t_{k+1}) + (\bar{\boldsymbol{x}}(t_{k+1}) - \bar{\boldsymbol{x}}(t_{k+1}))\,, t_{k+1}\right)\\
&= V\left(\bar{\boldsymbol{x}}(t_{k+1}) + (\boldsymbol{x}(t_{k+1}) - \bar{\boldsymbol{x}}(t_{k+1}))\,, t_{k+1}\right)\\
&= V(\bar{\boldsymbol{x}}(t_{k+1}) + \delta\boldsymbol{x}(t_{k+1}), t_{k+1})\\
&= V(\bar{\boldsymbol{x}}(t_{k+1}), t_{k+1}) + \nabla_{\boldsymbol{x}}V(\bar{\boldsymbol{x}}(t_{k+1}), t_{k+1})^T\delta\boldsymbol{x}(t_{k+1}) + \frac{1}{2}\delta\boldsymbol{x}(t_{k+1})^T\nabla_{\boldsymbol{xx}}V(\bar{\boldsymbol{x}}(t_{k+1}), t_{k+1})\delta\boldsymbol{x}(t_{k+1})
\end{aligned}$$

We denote:

$$\delta\boldsymbol{x}(t_{k+1}) = \boldsymbol{x}(t_{k+1}) - \bar{\boldsymbol{x}}(t_{k+1})$$
$$V(t_{k+1}) = V(\bar{\boldsymbol{x}}(t_{k+1}), t_{k+1})$$
$$V_{\boldsymbol{x}}(t_{k+1}) = \nabla_{\boldsymbol{x}}V(\bar{\boldsymbol{x}}(t_{k+1}), t_{k+1})$$
$$V_{\boldsymbol{xx}}(t_{k+1}) = \nabla_{\boldsymbol{xx}}V(\bar{\boldsymbol{x}}(t_{k+1}), t_{k+1})$$

Therefore:

$$= V(t_{k+1}) + V_{\boldsymbol{x}}(t_{k+1})^T\delta\boldsymbol{x}(t_{k+1}) + \frac{1}{2}\delta\boldsymbol{x}(t_{k+1})^TV_{\boldsymbol{xx}}(t_{k+1})\delta\boldsymbol{x}(t_{k+1})$$

We recall from the linearized dynamics in Part 1.1 that $\delta\boldsymbol{x}(t_{k+1}) = \boldsymbol{\Phi}(t_k)\delta\boldsymbol{x}(t_k)+\boldsymbol{B}(t_k)\delta\boldsymbol{u}(t_k)$. Therefore:

$$
\begin{aligned}
&= V(t_{k+1}) + V_{\boldsymbol{x}}(t_{k+1})^T\Big(\boldsymbol{\Phi}(t_k)\delta\boldsymbol{x}(t_k) + \boldsymbol{B}(t_k)\delta\boldsymbol{u}(t_k)\Big) \\
&+ \frac{1}{2}\Big(\boldsymbol{\Phi}(t_k)\delta\boldsymbol{x}(t_k) + \boldsymbol{B}(t_k)\delta\boldsymbol{u}(t_k)\Big)^T V_{\boldsymbol{xx}}(t_{k+1})\Big(\boldsymbol{\Phi}(t_k)\delta\boldsymbol{x}(t_k) + \boldsymbol{B}(t_k)\delta\boldsymbol{u}(t_k)\Big) \\
&= V(t_{k+1}) + V_{\boldsymbol{x}}(t_{k+1})^T\boldsymbol{\Phi}(t_k)\delta\boldsymbol{x}(t_k) + V_{\boldsymbol{x}}(t_{k+1})^T\boldsymbol{B}(t_k)\delta\boldsymbol{u}(t_k) \\
&+ \frac{1}{2}\delta\boldsymbol{x}(t_k)^T\boldsymbol{\Phi}(t_k)^T V_{\boldsymbol{xx}}(t_{k+1})\boldsymbol{\Phi}(t_k)\delta\boldsymbol{x}(t_k) + \frac{1}{2}\delta\boldsymbol{u}(t_k)^T\boldsymbol{B}(t_k)^T V_{\boldsymbol{xx}}(t_{k+1})\boldsymbol{B}(t_k)\delta\boldsymbol{u}(t_k) \\
&+ \frac{1}{2}\delta\boldsymbol{u}(t_k)^T\boldsymbol{B}(t_k)^T V_{\boldsymbol{xx}}(t_{k+1})\boldsymbol{\Phi}(t_k)\delta\boldsymbol{x}(t_k) + \frac{1}{2}\delta\boldsymbol{x}(t_k)^T\boldsymbol{\Phi}(t_k)^T V_{\boldsymbol{xx}}(t_{k+1})\boldsymbol{B}(t_k)\delta\boldsymbol{u}(t_k)
\end{aligned}
$$

The last two terms can be combined as they evaluate to equal scalars. Therefore, we can simplify the second order expansion of the value function as:

$$
\begin{aligned}
&V(\boldsymbol{x}(t_{k+1}), t_{k+1}) \\
&= V(t_{k+1}) + V_{\boldsymbol{x}}(t_{k+1})^T\boldsymbol{\Phi}(t_k)\delta\boldsymbol{x}(t_k) + V_{\boldsymbol{x}}(t_{k+1})^T\boldsymbol{B}(t_k)\delta\boldsymbol{u}(t_k) \\
&+ \frac{1}{2}\delta\boldsymbol{x}(t_k)^T\boldsymbol{\Phi}(t_k)^T V_{\boldsymbol{xx}}(t_{k+1})\boldsymbol{\Phi}(t_k)\delta\boldsymbol{x}(t_k) + \frac{1}{2}\delta\boldsymbol{u}(t_k)^T\boldsymbol{B}(t_k)^T V_{\boldsymbol{xx}}(t_{k+1})\boldsymbol{B}(t_k)\delta\boldsymbol{u}(t_k) \\
&+ \delta\boldsymbol{u}(t_k)^T\boldsymbol{B}(t_k)^T V_{\boldsymbol{xx}}(t_{k+1})\boldsymbol{\Phi}(t_k)\delta\boldsymbol{x}(t_k)
\end{aligned}
$$

We can now consider the second order expansion of the action-value function itself:

$$
\begin{aligned}
&Q(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k), t_k) \\
&= Q(\boldsymbol{x}(t_k) + (\bar{\boldsymbol{x}}(t_k) - \bar{\boldsymbol{x}}(t_k)), \boldsymbol{u}(t_k) + (\bar{\boldsymbol{u}}(t_k) - \bar{\boldsymbol{u}}(t_k)), t_k) \\
&= Q(\bar{\boldsymbol{x}}(t_k) + (\boldsymbol{x}(t_k) - \bar{\boldsymbol{x}}(t_k)), \bar{\boldsymbol{u}}(t_k) + (\boldsymbol{u}(t_k) - \bar{\boldsymbol{u}}(t_k)), t_k) \\
&= Q(\bar{\boldsymbol{x}}(t_k) + \delta\boldsymbol{x}(t_k), \bar{\boldsymbol{u}}(t_k) + \delta\boldsymbol{u}(t_k), t_k) \\
&= Q_0 + Q_{\boldsymbol{x}}^T\delta\boldsymbol{x}(t_k) + Q_{\boldsymbol{u}}^T\delta\boldsymbol{u}(t_k) + \frac{1}{2}\delta\boldsymbol{x}(t_k)^T Q_{\boldsymbol{xx}}\delta\boldsymbol{x}(t_k) + \frac{1}{2}\delta\boldsymbol{u}(t_k)^T Q_{\boldsymbol{uu}}\delta\boldsymbol{u}(t_k) \\
&+ \frac{1}{2}\delta\boldsymbol{u}(t_k)^T Q_{\boldsymbol{xu}}\delta\boldsymbol{x}(t_k) + \frac{1}{2}\delta\boldsymbol{x}(t_k)^T Q_{\boldsymbol{ux}}\delta\boldsymbol{u}(t_k)
\end{aligned}
$$

Similiarly to the value function expansion, the last two terms can be combined as they evaluate to equal scalars, we can simplify the second order expansion of the state-action value function as:

$$
\begin{aligned}
&Q(\bar{\boldsymbol{x}}(t_k) + \delta\boldsymbol{x}(t_k), \bar{\boldsymbol{u}}(t_k) + \delta\boldsymbol{u}(t_k), t_k) \\
&= Q_0 + Q_{\boldsymbol{x}}^T\delta\boldsymbol{x}(t_k) + Q_{\boldsymbol{u}}^T\delta\boldsymbol{u}(t_k) + \frac{1}{2}\delta\boldsymbol{x}(t_k)^T Q_{\boldsymbol{xx}}\delta\boldsymbol{x}(t_k) + \frac{1}{2}\delta\boldsymbol{u}(t_k)^T Q_{\boldsymbol{uu}}\delta\boldsymbol{u}(t_k) \\
&+ \delta\boldsymbol{x}(t_k)^T Q_{\boldsymbol{ux}}\delta\boldsymbol{u}(t_k)
\end{aligned}
$$

Finally we can combine the second order expansions of the running cost and value function to express the second order expansion of the state-action value function:

$$
\begin{aligned}
&Q(\bar{\boldsymbol{x}}(t_k)+\delta\boldsymbol{x}(t_k),\bar{\boldsymbol{u}}(t_k)+\delta\boldsymbol{u}(t_k),t_k)\\
&=\mathcal{L}+\mathcal{L}_{\boldsymbol{x}}^T\delta\boldsymbol{x}(t_k)+\mathcal{L}_{\boldsymbol{u}}^T\delta\boldsymbol{u}(t_k)+\frac{1}{2}\delta\boldsymbol{x}(t_k)^T\mathcal{L}_{\boldsymbol{xx}}\delta\boldsymbol{x}(t_k)+\frac{1}{2}\delta\boldsymbol{u}(t_k)^T\mathcal{L}_{\boldsymbol{uu}}\delta\boldsymbol{u}(t_k)\\
&+\frac{1}{2}\delta\boldsymbol{u}(t_k)^T\mathcal{L}_{\boldsymbol{xu}}\delta\boldsymbol{x}(t_k)+\frac{1}{2}\delta\boldsymbol{x}(t_k)^T\mathcal{L}_{\boldsymbol{ux}}\delta\boldsymbol{u}(t_k)\\
&+V(t_{k+1})+V_{\boldsymbol{x}}(t_{k+1})^T\boldsymbol{\Phi}(t_k)\delta\boldsymbol{x}(t_k)+V_{\boldsymbol{x}}(t_{k+1})^T\boldsymbol{B}(t_k)\delta\boldsymbol{u}(t_k)\\
&+\frac{1}{2}\delta\boldsymbol{x}(t_k)^T\boldsymbol{\Phi}(t_k)^TV_{\boldsymbol{xx}}(t_{k+1})\boldsymbol{\Phi}(t_k)\delta\boldsymbol{x}(t_k)+\frac{1}{2}\delta\boldsymbol{u}(t_k)^T\boldsymbol{B}(t_k)^TV_{\boldsymbol{xx}}(t_{k+1})\boldsymbol{B}(t_k)\delta\boldsymbol{u}(t_k)\\
&+\delta\boldsymbol{u}(t_k)^T\boldsymbol{B}(t_k)^TV_{\boldsymbol{xx}}(t_{k+1})\boldsymbol{\Phi}(t_k)\delta\boldsymbol{x}(t_k)
\end{aligned}
$$

We can group the zero, first, and second order terms as:

$$
\begin{aligned}
&=\underbrace{\Big(\mathcal{L}+V(t_{k+1})\Big)}_{Q_0}+\underbrace{\Big(\mathcal{L}_{\boldsymbol{x}}+\boldsymbol{\Phi}(t_k)^TV_{\boldsymbol{x}}(t_{k+1})\Big)^T}_{Q_{\boldsymbol{x}}}\delta\boldsymbol{x}(t_k)+\underbrace{\Big(\mathcal{L}_{\boldsymbol{u}}+\boldsymbol{B}(t_k)^TV_{\boldsymbol{x}}(t_{k+1})\Big)^T}_{Q_{\boldsymbol{u}}}\delta\boldsymbol{u}(t_k)\\
&+\frac{1}{2}\delta\boldsymbol{x}(t_k)^T\underbrace{\Big(\mathcal{L}_{\boldsymbol{xx}}+\boldsymbol{\Phi}(t_k)^TV_{\boldsymbol{xx}}(t_{k+1})\boldsymbol{\Phi}(t_k)\Big)}_{Q_{\boldsymbol{xx}}}\delta\boldsymbol{x}(t_k)\\
&+\frac{1}{2}\delta\boldsymbol{u}(t_k)^T\underbrace{\Big(\mathcal{L}_{\boldsymbol{uu}}+\boldsymbol{B}(t_k)^TV_{\boldsymbol{xx}}(t_{k+1})\boldsymbol{B}(t_k)\Big)}_{Q_{\boldsymbol{uu}}}\delta\boldsymbol{u}(t_k)\\
&+\delta\boldsymbol{u}(t_k)^T\underbrace{\left(\frac{1}{2}\mathcal{L}_{\boldsymbol{ux}}+\frac{1}{2}\mathcal{L}_{\boldsymbol{xu}}^T+\frac{1}{2}\boldsymbol{B}(t_k)^T\Big(V_{\boldsymbol{xx}}(t_{k+1})+V_{\boldsymbol{xx}}(t_{k+1})^T\Big)\boldsymbol{\Phi}(t_k)\right)}_{Q_{\boldsymbol{ux}}}\delta\boldsymbol{x}(t_k)
\end{aligned}
$$

Where we denote:

$$
\begin{aligned}
Q_0&=\mathcal{L}+V(t_{k+1})\\
Q_{\boldsymbol{x}}&=\mathcal{L}_{\boldsymbol{x}}+\boldsymbol{\Phi}(t_k)^TV_{\boldsymbol{x}}(t_{k+1})\\
Q_{\boldsymbol{u}}&=\mathcal{L}_{\boldsymbol{u}}+\boldsymbol{B}(t_k)^TV_{\boldsymbol{x}}(t_{k+1})\\
Q_{\boldsymbol{xx}}&=\mathcal{L}_{\boldsymbol{xx}}+\boldsymbol{\Phi}(t_k)^TV_{\boldsymbol{xx}}(t_{k+1})\boldsymbol{\Phi}(t_k)\\
Q_{\boldsymbol{uu}}&=\mathcal{L}_{\boldsymbol{uu}}+\boldsymbol{B}(t_k)^TV_{\boldsymbol{xx}}(t_{k+1})\boldsymbol{B}(t_k)\\
Q_{\boldsymbol{ux}}&=\boldsymbol{B}(t_k)^TV_{\boldsymbol{xx}}(t_{k+1})\boldsymbol{\Phi}(t_k)+\mathcal{L}_{\boldsymbol{ux}}
\end{aligned}
$$

1.3) Next we compute the optimal control corrections $\delta\boldsymbol{u}^*(t_k)$. We do this by minimizing $\delta\boldsymbol{u}(t_k)$ with respect to the state-action value function $Q(\boldsymbol{x}(t_k),\boldsymbol{u}(t_k))$. This is the same as minimizing $\boldsymbol{u}(t_k)$ to compute the locally optimal control $\boldsymbol{u}^*(t_k)$ in the Bellman Principle. This so since $\boldsymbol{u}^*(t_k)=\bar{\boldsymbol{u}}(t_k)+\delta\boldsymbol{u}^*(t_k)$.

$$\min_{\delta\boldsymbol{u}(t_k)}\Bigg[Q(\boldsymbol{x}(t_k),\boldsymbol{u}(t_k))\Bigg]$$

$$=\min_{\delta\boldsymbol{u}(t_k)}\Bigg[\Big(\mathcal{L}+V(t_{k+1})\Big)+\Big(\mathcal{L}_{\boldsymbol{x}}+\boldsymbol{\Phi}(t_k)^TV_{\boldsymbol{x}}(t_{k+1})\Big)^T\delta\boldsymbol{x}(t_k)+\Big(\mathcal{L}_{\boldsymbol{u}}+\boldsymbol{B}(t_k)^TV_{\boldsymbol{x}}(t_{k+1})\Big)^T\delta\boldsymbol{u}(t_k)$$

$$+\frac{1}{2}\delta\boldsymbol{x}(t_k)^T\Big(\mathcal{L}_{\boldsymbol{xx}}+\boldsymbol{\Phi}(t_k)^TV_{\boldsymbol{xx}}(t_{k+1})\boldsymbol{\Phi}(t_k)\Big)\delta\boldsymbol{x}(t_k)+\frac{1}{2}\delta\boldsymbol{u}(t_k)^T\Big(\mathcal{L}_{\boldsymbol{uu}}+\boldsymbol{B}(t_k)^TV_{\boldsymbol{xx}}(t_{k+1})\boldsymbol{B}(t_k)\Big)\delta\boldsymbol{u}(t_k)$$

$$+\delta\boldsymbol{u}(t_k)^T\left(\frac{1}{2}\mathcal{L}_{\boldsymbol{ux}}+\frac{1}{2}\mathcal{L}_{\boldsymbol{xu}}^T+\frac{1}{2}\boldsymbol{B}(t_k)^T\Big(V_{\boldsymbol{xx}}(t_{k+1})+V_{\boldsymbol{xx}}(t_{k+1})^T\Big)\boldsymbol{\Phi}(t_k)\right)\delta\boldsymbol{x}(t_k)\Bigg]$$

To solve this take the gradient of $Q(\boldsymbol{x}(t_k),\boldsymbol{u}(t_k))$ with respect to $\delta\boldsymbol{u}(t_k)$ and set it equal to zero:

$$0=\underbrace{\Big(\mathcal{L}_{\boldsymbol{u}}+\boldsymbol{B}(t_k)^TV_{\boldsymbol{x}}(t_{k+1})\Big)}_{Q_{\boldsymbol{u}}}+\underbrace{\left(\frac{1}{2}\mathcal{L}_{\boldsymbol{ux}}+\frac{1}{2}\mathcal{L}_{\boldsymbol{xu}}^T+\frac{1}{2}\boldsymbol{B}(t_k)^T\Big(V_{\boldsymbol{xx}}(t_{k+1})+V_{\boldsymbol{xx}}(t_{k+1})^T\Big)\boldsymbol{\Phi}(t_k)\right)}_{Q_{\boldsymbol{ux}}}\delta\boldsymbol{x}(t_k)$$

$$+\underbrace{\Big(\mathcal{L}_{\boldsymbol{uu}}+\boldsymbol{B}(t_k)^TV_{\boldsymbol{xx}}(t_{k+1})\boldsymbol{B}(t_k)\Big)}_{Q_{\boldsymbol{uu}}}\delta\boldsymbol{u}(t_k)$$

Therefore:

$$\begin{aligned}
0&=Q_{\boldsymbol{u}}+Q_{\boldsymbol{ux}}\delta\boldsymbol{x}(t_k)+Q_{\boldsymbol{uu}}\delta\boldsymbol{u}(t_k)\\
-Q_{\boldsymbol{uu}}\delta\boldsymbol{u}(t_k)&=Q_{\boldsymbol{u}}+Q_{\boldsymbol{ux}}\delta\boldsymbol{x}(t_k)\\
\delta\boldsymbol{u}(t_k)&=\underbrace{-Q_{\boldsymbol{uu}}^{-1}Q_{\boldsymbol{u}}}_{\boldsymbol{l}_{\boldsymbol{u}}(t_k)}+\underbrace{\Big(-Q_{\boldsymbol{uu}}^{-1}Q_{\boldsymbol{ux}}\Big)}_{\boldsymbol{L}_{\boldsymbol{u}}(t_k)}\delta\boldsymbol{x}(t_k)
\end{aligned}$$

Further, we denote $\boldsymbol{l}_{\boldsymbol{u}}(t_k)$ as the feedforward term and $\boldsymbol{L}_{\boldsymbol{u}}(t_k)$ as the feedback term:

$$\boldsymbol{l}_{\boldsymbol{u}}(t_k)=-Q_{\boldsymbol{uu}}^{-1}Q_{\boldsymbol{u}}$$

$$\boldsymbol{L}_{\boldsymbol{u}}(t_k)=-Q_{\boldsymbol{uu}}^{-1}Q_{\boldsymbol{ux}}$$

And determine the optimal control corrections to be:

$$\delta\boldsymbol{u}^*(t_k)=\boldsymbol{l}_{\boldsymbol{u}}(t_k)+\boldsymbol{L}_{\boldsymbol{u}}(t_k)\delta\boldsymbol{x}(t_k)$$

1.4) We can compute the equations $V(t_k), V_{\boldsymbol{x}}(t_k), V_{\boldsymbol{xx}}(t_k)$ in backward time by substituting in our optimal control correction $\delta\boldsymbol{u}^*(t_k)$ into the expanded state-action value function $Q(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k))$.

$$
\begin{aligned}
&Q(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k))\Big|_{\delta\boldsymbol{u}^*(t_k)=\boldsymbol{l_u}(t_k)+\boldsymbol{L_u}(t_k)\delta\boldsymbol{x}(t_k)} \\
&= Q_0 + Q_{\boldsymbol{x}}^T\delta\boldsymbol{x}(t_k) + Q_{\boldsymbol{u}}^T\Big(\boldsymbol{l_u}(t_k) + \boldsymbol{L_u}(t_k)\delta\boldsymbol{x}(t_k)\Big) \\
&+ \frac{1}{2}\delta\boldsymbol{x}(t_k)^T Q_{\boldsymbol{xx}}\delta\boldsymbol{x}(t_k) + \frac{1}{2}\Big(\boldsymbol{l_u}(t_k) + \boldsymbol{L_u}(t_k)\delta\boldsymbol{x}(t_k)\Big)^T Q_{\boldsymbol{uu}}\Big(\boldsymbol{l_u}(t_k) + \boldsymbol{L_u}(t_k)\delta\boldsymbol{x}(t_k)\Big) \\
&+ \frac{1}{2}\Big(\boldsymbol{l_u}(t_k) + \boldsymbol{L_u}(t_k)\delta\boldsymbol{x}(t_k)\Big)^T Q_{\boldsymbol{ux}}\delta\boldsymbol{x}(t_k) + \frac{1}{2}\delta\boldsymbol{x}(t_k)^T Q_{\boldsymbol{xu}}\Big(\boldsymbol{l_u}(t_k) + \boldsymbol{L_u}(t_k)\delta\boldsymbol{x}(t_k)\Big)
\end{aligned}
$$

To compute the backward terms we can denote:

$$V(\bar{\boldsymbol{x}}(t_k) + \delta\boldsymbol{x}(t_k), t) = V(t_k) + V_{\boldsymbol{x}}(t_k)^T\delta\boldsymbol{x}(t_k) + \frac{1}{2}\delta\boldsymbol{x}(t_k)^T V_{\boldsymbol{xx}}(t_k)\delta\boldsymbol{x}(t_k)$$

Therefore the zero order terms of the state-action value function in $\delta\boldsymbol{x}$ are equal to $V(t_k)$, the first order terms are equal to $V_{\boldsymbol{x}}(t_k)$, and the second order terms are equal to $V_{\boldsymbol{xx}}(t_k)$.

First can group the zero order terms that equate to $V(t_k)$:

$$V(t_k) = Q_0 + Q_{\boldsymbol{u}}^T\boldsymbol{l_u}(t_k) + \frac{1}{2}\boldsymbol{l_u}(t_k)^T Q_{\boldsymbol{uu}}\boldsymbol{l_u}(t_k)$$

Next we can group the first order terms that equate to $V_{\boldsymbol{x}}(t_k)$:

$$V_{\boldsymbol{x}}(t_k) = Q_{\boldsymbol{x}} + Q_{\boldsymbol{u}}^T\boldsymbol{L_u}(t_k) + Q_{\boldsymbol{xu}}\boldsymbol{l_u}(t_k) + \boldsymbol{L_u}(t_k)^T Q_{\boldsymbol{uu}}\boldsymbol{l_u}(t_k)$$

Finally we can group the first order terms that equate to $V_{\boldsymbol{xx}}(t_k)$:

$$V_{\boldsymbol{xx}}(t_k) = Q_{\boldsymbol{xx}} + \boldsymbol{L_u}(t_k)^T Q_{\boldsymbol{uu}}\boldsymbol{L_u}(t_k) + \boldsymbol{L_u}(t_k)^T Q_{\boldsymbol{ux}} + Q_{\boldsymbol{xu}}\boldsymbol{L_u}(t_k)$$

In summary, the backward equations can be expressed as:

$$
\begin{aligned}
V(t_k) &= Q_0 + Q_{\boldsymbol{u}}^T\boldsymbol{l_u}(t_k) + \frac{1}{2}\boldsymbol{l_u}(t_k)^T Q_{\boldsymbol{uu}}\boldsymbol{l_u}(t_k) \\
V_{\boldsymbol{x}}(t_k) &= Q_{\boldsymbol{x}} + Q_{\boldsymbol{u}}^T\boldsymbol{L_u}(t_k) + Q_{\boldsymbol{xu}}\boldsymbol{l_u}(t_k) + \boldsymbol{L_u}(t_k)^T Q_{\boldsymbol{uu}}\boldsymbol{l_u}(t_k) \\
V_{\boldsymbol{xx}}(t_k) &= Q_{\boldsymbol{xx}} + \boldsymbol{L_u}(t_k)^T Q_{\boldsymbol{uu}}\boldsymbol{L_u}(t_k) + \boldsymbol{L_u}(t_k)^T Q_{\boldsymbol{ux}} + Q_{\boldsymbol{xu}}\boldsymbol{L_u}(t_k)
\end{aligned}
$$

Further we can express these simply as functions of state-action value function by substituting in $\boldsymbol{l_u}(t_k)$ and $\boldsymbol{L_u}(t_k)$:

We do this for the value function:

$$V(t_k) = Q_0 + Q_{\boldsymbol{u}}^T\Big(-Q_{\boldsymbol{uu}}^{-1}Q_{\boldsymbol{u}}\Big) + \frac{1}{2}\Big(-Q_{\boldsymbol{uu}}^{-1}Q_{\boldsymbol{u}}\Big)^T Q_{\boldsymbol{uu}}\Big(-Q_{\boldsymbol{uu}}^{-1}Q_{\boldsymbol{u}}\Big)$$

$$= Q_0 - Q_{\boldsymbol{u}}^T Q_{\boldsymbol{uu}}^{-1} Q_{\boldsymbol{u}} + \frac{1}{2} Q_{\boldsymbol{u}}^T Q_{\boldsymbol{uu}}^{-1} \underbrace{Q_{\boldsymbol{uu}} Q_{\boldsymbol{uu}}^{-1}}_{\boldsymbol{I}} Q_{\boldsymbol{u}}$$

$$= Q_0 - Q_{\boldsymbol{u}}^T Q_{\boldsymbol{uu}}^{-1} Q_{\boldsymbol{u}} + \frac{1}{2} Q_{\boldsymbol{u}}^T Q_{\boldsymbol{uu}}^{-1} Q_{\boldsymbol{u}}$$

$$= Q_0 - \frac{1}{2} Q_{\boldsymbol{u}}^T Q_{\boldsymbol{uu}}^{-1} Q_{\boldsymbol{u}}$$

Then, we do this for the gradient of the value function:

$$V_{\boldsymbol{x}}(t_k) = Q_{\boldsymbol{x}} + Q_{\boldsymbol{u}}^T\Big(-Q_{\boldsymbol{uu}}^{-1}Q_{\boldsymbol{ux}}\Big) + Q_{\boldsymbol{xu}}\Big(-Q_{\boldsymbol{uu}}^{-1}Q_{\boldsymbol{u}}\Big) + \Big(-Q_{\boldsymbol{uu}}^{-1}Q_{\boldsymbol{ux}}\Big)^T Q_{\boldsymbol{uu}}\Big(-Q_{\boldsymbol{uu}}^{-1}Q_{\boldsymbol{u}}\Big)$$

$$= Q_{\boldsymbol{x}} - Q_{\boldsymbol{u}}^T Q_{\boldsymbol{uu}}^{-1} Q_{\boldsymbol{ux}} - Q_{\boldsymbol{xu}} Q_{\boldsymbol{uu}}^{-1} Q_{\boldsymbol{u}} + Q_{\boldsymbol{ux}}^T Q_{\boldsymbol{uu}}^{-1} \underbrace{Q_{\boldsymbol{uu}} Q_{\boldsymbol{uu}}^{-1}}_{\boldsymbol{I}} Q_{\boldsymbol{u}}$$

$$= Q_{\boldsymbol{x}} - Q_{\boldsymbol{u}}^T Q_{\boldsymbol{uu}}^{-1} Q_{\boldsymbol{ux}} - Q_{\boldsymbol{xu}} Q_{\boldsymbol{uu}}^{-1} Q_{\boldsymbol{u}} + Q_{\boldsymbol{u}}^T Q_{\boldsymbol{uu}}^{-1} Q_{\boldsymbol{ux}}$$

$$= Q_{\boldsymbol{x}} - Q_{\boldsymbol{xu}} Q_{\boldsymbol{uu}}^{-1} Q_{\boldsymbol{u}}$$

Finally, we do this for the Hessian of the value function:

$$V_{\boldsymbol{xx}}(t_k) = Q_{\boldsymbol{xx}} + \boldsymbol{L_u}(t_k)^T Q_{\boldsymbol{uu}} \boldsymbol{L_u}(t_k) + \boldsymbol{L_u}(t_k)^T Q_{\boldsymbol{ux}} + Q_{\boldsymbol{xu}} \boldsymbol{L_u}(t_k)$$

$$= Q_{\boldsymbol{xx}} + \Big(-Q_{\boldsymbol{uu}}^{-1}Q_{\boldsymbol{ux}}\Big)^T Q_{\boldsymbol{uu}}\Big(-Q_{\boldsymbol{uu}}^{-1}Q_{\boldsymbol{ux}}\Big) + \Big(-Q_{\boldsymbol{uu}}^{-1}Q_{\boldsymbol{ux}}\Big)^T Q_{\boldsymbol{ux}} + Q_{\boldsymbol{xu}}\Big(-Q_{\boldsymbol{uu}}^{-1}Q_{\boldsymbol{ux}}\Big)$$

$$= Q_{\boldsymbol{xx}} + Q_{\boldsymbol{ux}}^T Q_{\boldsymbol{uu}}^{-1} \underbrace{Q_{\boldsymbol{uu}} Q_{\boldsymbol{uu}}^{-1}}_{\boldsymbol{I}} Q_{\boldsymbol{ux}} - Q_{\boldsymbol{ux}}^T Q_{\boldsymbol{uu}}^{-1} Q_{\boldsymbol{ux}} - Q_{\boldsymbol{xu}} Q_{\boldsymbol{uu}}^{-1} Q_{\boldsymbol{ux}}$$

$$= Q_{\boldsymbol{xx}} + Q_{\boldsymbol{ux}}^T Q_{\boldsymbol{uu}}^{-1} Q_{\boldsymbol{ux}} - Q_{\boldsymbol{ux}}^T Q_{\boldsymbol{uu}}^{-1} Q_{\boldsymbol{ux}} - Q_{\boldsymbol{xu}} Q_{\boldsymbol{uu}}^{-1} Q_{\boldsymbol{ux}}$$

$$= Q_{\boldsymbol{xx}} - Q_{\boldsymbol{xu}} Q_{\boldsymbol{uu}}^{-1} Q_{\boldsymbol{ux}}$$

In summary, the backward equations can be expressed as functions of the state-action value function as:

$$V(t_k) = Q_0 - \frac{1}{2} Q_{\boldsymbol{u}}^T Q_{\boldsymbol{uu}}^{-1} Q_{\boldsymbol{u}}$$

$$V_{\boldsymbol{x}}(t_k) = Q_{\boldsymbol{x}} - Q_{\boldsymbol{xu}} Q_{\boldsymbol{uu}}^{-1} Q_{\boldsymbol{u}}$$

$$V_{\boldsymbol{xx}}(t_k) = Q_{\boldsymbol{xx}} - Q_{\boldsymbol{xu}} Q_{\boldsymbol{uu}}^{-1} Q_{\boldsymbol{ux}}$$

Part 2.

2.1) See MATLAB code implementation.

To run this simulation run the Inverted_Pendulum/main.m file.

2.2) See MATLAB code implementation.

To run this simulation run the CartPole/main.m file.

2.3) See MATLAB code implementation.

To run this simulation run the Inverted_Pendulum/main_robustness_test.m file.

In this problem we are asked to test the robustness of our DDP policy against stochastic forces that act as disturbances in our dynamics.

$$I\ddot{\theta} + b\dot{\theta} + mglsin(\theta) = u + f_{stochastic}$$

We do this by running the DDP optimization until it converges. This results in an optimal trajectory $\boldsymbol{x}^*(t_k)$, optimal control input $\boldsymbol{u}^*(t_k)$, and optimal feeback gains $\boldsymbol{L}^*(t_k)$. We then initialize a new trajectory and propagate the dynamics forward, perturbed by a stochastic force term, using the following controller: $\boldsymbol{u}(t_k) = \boldsymbol{u}^*(t_k) + \boldsymbol{L}^*(t_k)\big(\boldsymbol{x}(t_k) - \boldsymbol{x}^*(t_k)\big)$

This process is repeated for many trajectories throught the following algorithm:

Algorithm 1: DDP Robustness Test Against Stochastic Forces

Initialize: $\boldsymbol{x}^*(t_k), \boldsymbol{u}^*(t_k), \boldsymbol{L}^*(t_k)$;
Initialize: $\boldsymbol{H}$: time horizon, dt :time step, $\boldsymbol{num\_traj}$: number of trajectories;
Initialize: $\boldsymbol{F_x}(\boldsymbol{x}(t_k)), \boldsymbol{F_u}(\boldsymbol{u}(t_k))$: dynamics;
Initialize: $\boldsymbol{x}(t_k) = \boldsymbol{0}$;
Initialize: σ: noise level;
for $\boldsymbol{i} = 1$ *to* $\boldsymbol{num\_traj}$ **do**
  for $\boldsymbol{k} = 1$ *to* $\boldsymbol{H} - 1$ **do**
    $R = \sigma \times randn$;
    $\boldsymbol{u}(t_k) = \boldsymbol{u}^*(t_k) + \boldsymbol{L}^*(t_k)\big(\boldsymbol{x}(t_k) - \boldsymbol{x}^*(t_k)\big)$;
    $\boldsymbol{x}(t_{k+1}) = \boldsymbol{F_x}(\boldsymbol{x}(t_k))dt + \boldsymbol{F_u}(\boldsymbol{u}(t_k))dt + \boldsymbol{F_u}(\boldsymbol{u}(t_k))R\sqrt{dt}$
  end
end

The following plot shows the results of DDP after optimization:

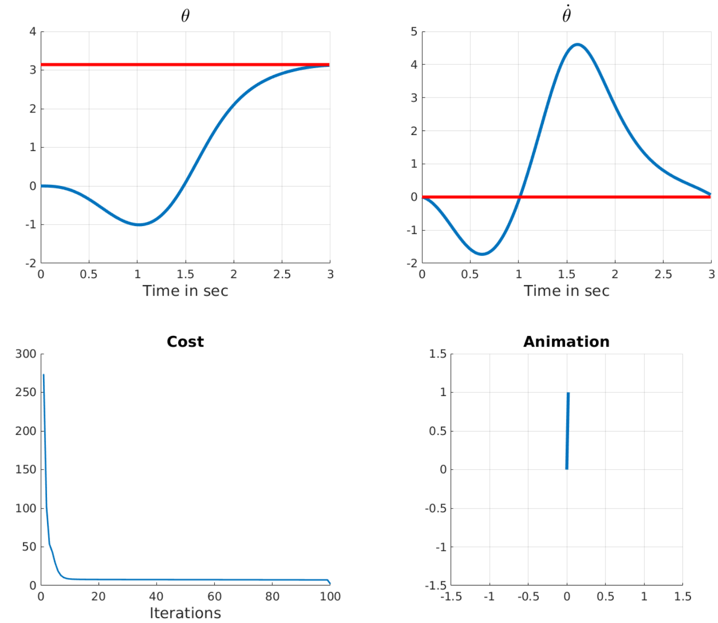

Figure 1: DDP optimal results.

The following plot shows the results of running the Robustness Test algorithm for 20 trajectories using the optimal DDP trajectory, controller, and feedback gains with a noise level of $\sigma = 1.0$:

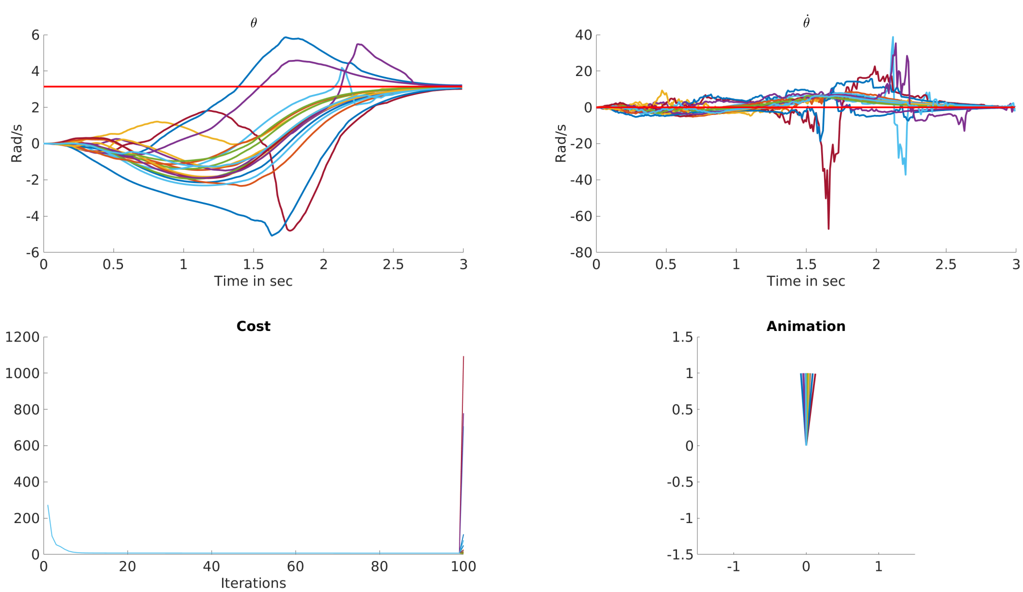

Figure 2: Robustness Test results.