# Spacecraft Formation Flying using differential dynamic programming

Tomohiro Sasaki

Nov 29th, 2022

Georgia Tech

Daniel Guggenheim School of Aerospace Engineering

SSDL | Space Systems Design Laboratory

# About Me

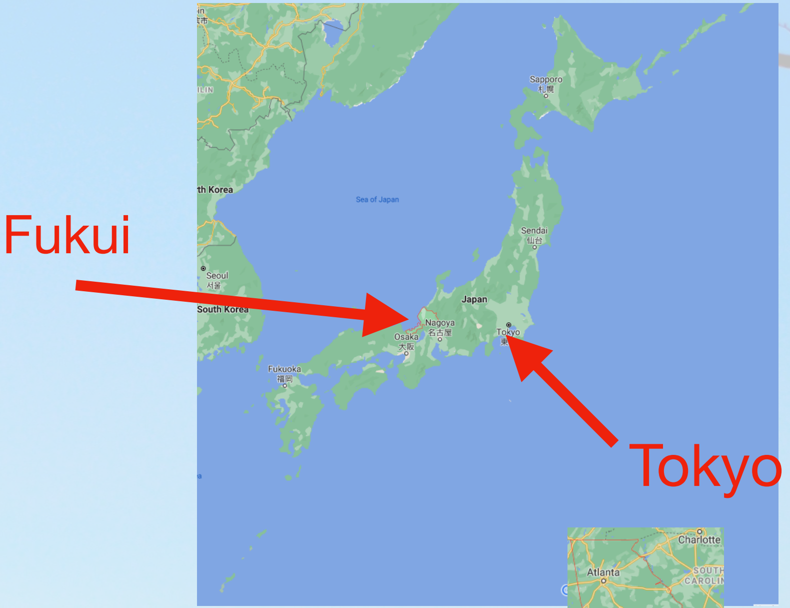
Name

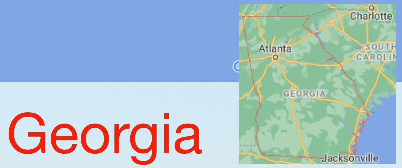
- Sasaki, Tomohiro(佐々木, 智宏)

- Origin
  - Born and raised in Fukui, Japan
- In my free time
  - powerlifting
- Bachelor's Degree
  - B.E. in AE from the Tokyo Metropolitan University in Tokyo, Japan
  - Research topic: RF Cathode for Low-power Hall Thruster"

Current interest

- High-performance optimal control
- Stochastic optimal control -> master's problem
- Motion planning under uncertainty -> dissertation




# World Cup Group E table

| | Team | PTS | GP | W | L | D | GF | GA | GD |
|---|---|---|---|---|---|---|---|---|---|
| 🇪🇸 | 1. Spain | 4 | 2 | 1 | 0 | 1 | 8 | 1 | +7 |
| 🇯🇵 | 2. Japan | 3 | 2 | 1 | 1 | 0 | 2 | 2 | 0 |
| 🇨🇷 | 3. Costa Rica | 3 | 2 | 1 | 1 | 0 | 1 | 7 | -6 |
| 🇩🇪 | 4. Germany | 1 | 2 | 0 | 1 | 1 | 2 | 3 | -1 |

# Outline

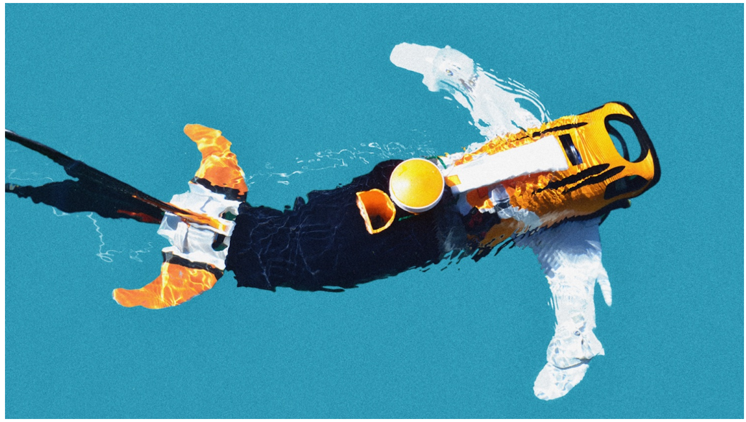
- Motivation
- Differential Dynamic Programming (DDP)
- Constrained DDP
- Dynamics and Constraints
- Numerical Simulations
- Future Work and Current Progress
- Conclusion

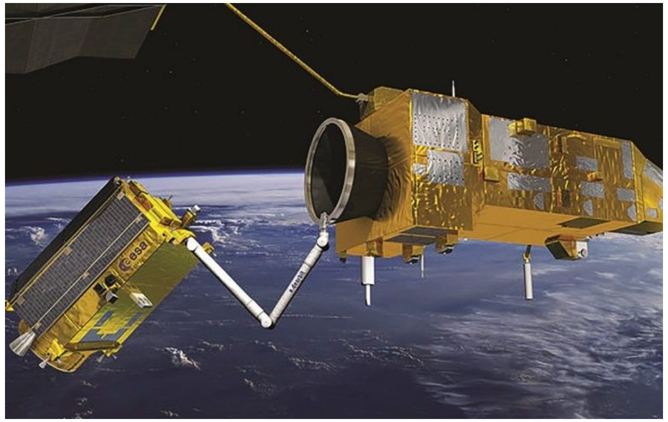
# Outline

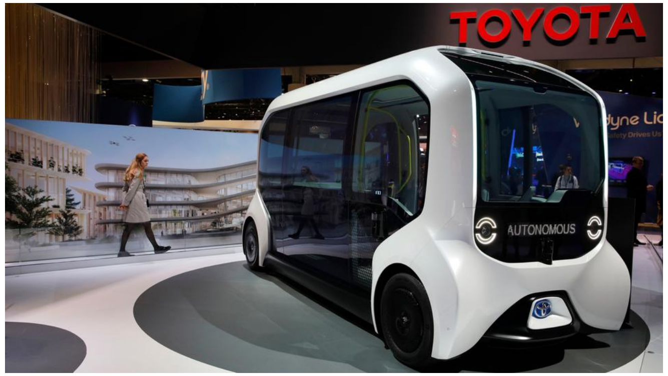
- Motivation
- Differential Dynamic Programming (DDP)
- Constrained DDP
- Dynamics and Constraints
- Numerical Simulations
- Future Work and Current Progress
- Conclusion

# Motivation (1/3)

- Autonomous motion planning and control is an active research area for future intelligent robotic systems
  - spacecraft formation flying
  - self-driving car
  - autonomous underwater vehicle (AUV)
  - unmanned aerial vehicle (UAV)

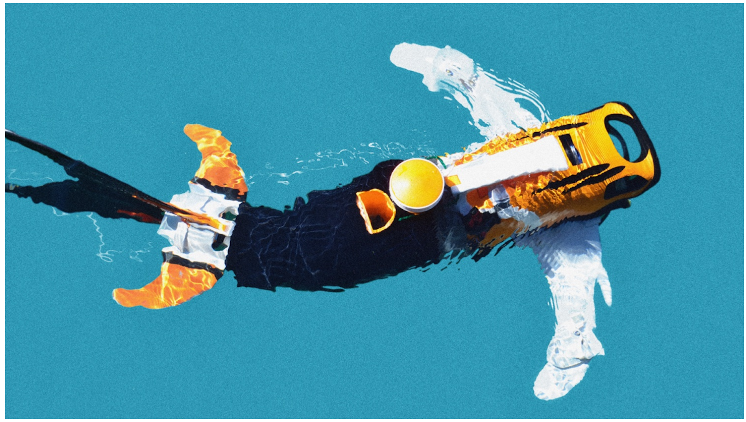


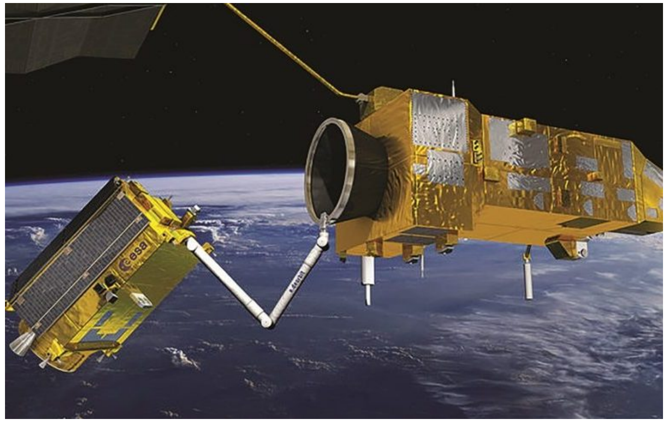


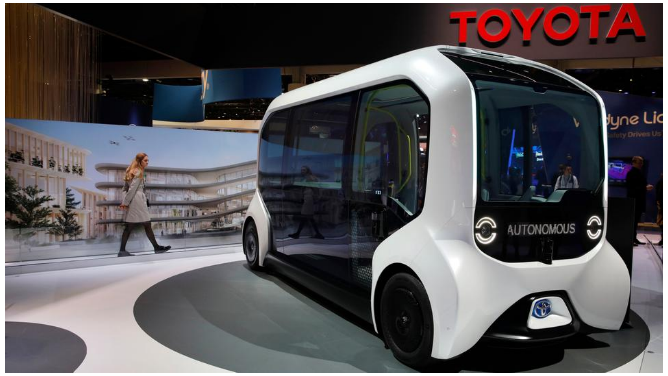


# Motivation (1/3)

- Autonomous motion planning and control is an active research area for future intelligent robotic systems
  - spacecraft formation flying
  - self-driving car
  - autonomous underwater vehicle (AUV)
  - unmanned aerial vehicle (UAV)
- Although optimal control was born in 1697 and established in 1956, its capability is not enough as the application becomes complex





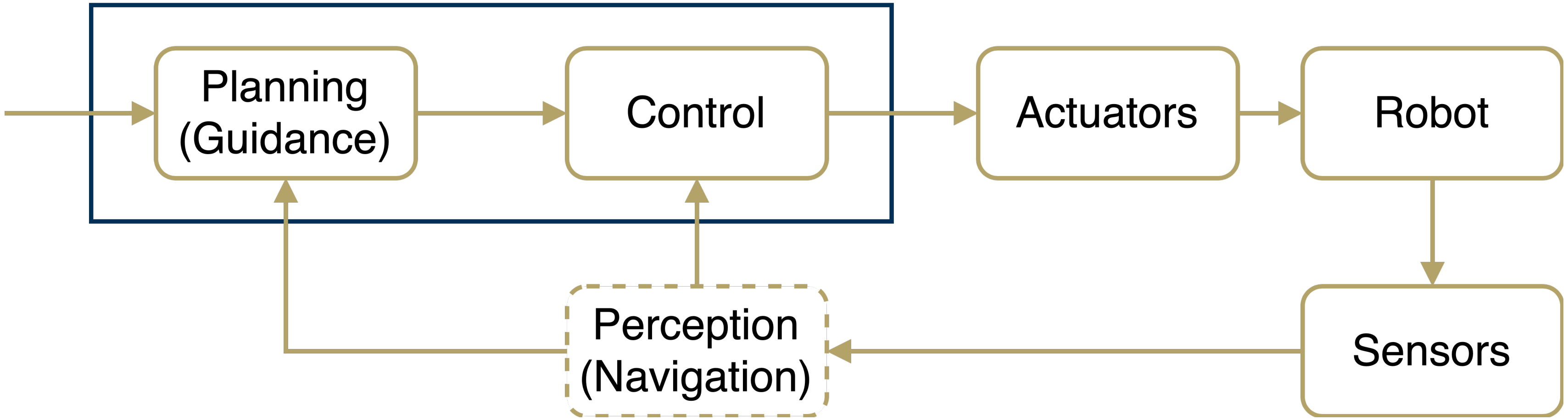


# Motivation (2/3)

- **Planning**: providing a “good” nominal trajectory to track -> **Open-loop**
- **Control**: providing 1) stability, 2) tracking, 3) disturbance rejection, 4) robustness -> **Closed-loop**

# Motivation (2/3)

- **Planning**: providing a "good" nominal trajectory to track -> **Open-loop**
- **Control**: providing 1) stability, 2) tracking, 3) disturbance rejection, 4) robustness -> **Closed-loop**
- General perception, planning, and control (or guidance, navigation, and control; GNC) flow



# Motivation (3/3)

- Usually, planning and control algorithms are not identically chosen
  - i.e. choose tree search for planning and LQR for control
  - planning algorithm may not provide feedback
  - control algorithm may not be efficient by itself

# Motivation (3/3)

- Usually, planning and control algorithms are not identically chosen
  - i.e. choose tree search for planning and LQR for control
  - planning algorithm may not provide feedback
  - control algorithm may not be efficient by itself
- Ultimately, we want to find the best combination of these algorithms
- Or, we want to develop an algorithm that completes both tasks at the same time.

# Contribution

- Developing an algorithm that solves planning and control tasks under arbitrary constraints at the same time and ultimately provides robustness to the system even in an uncertain environment

# Optimal Control (1/2)

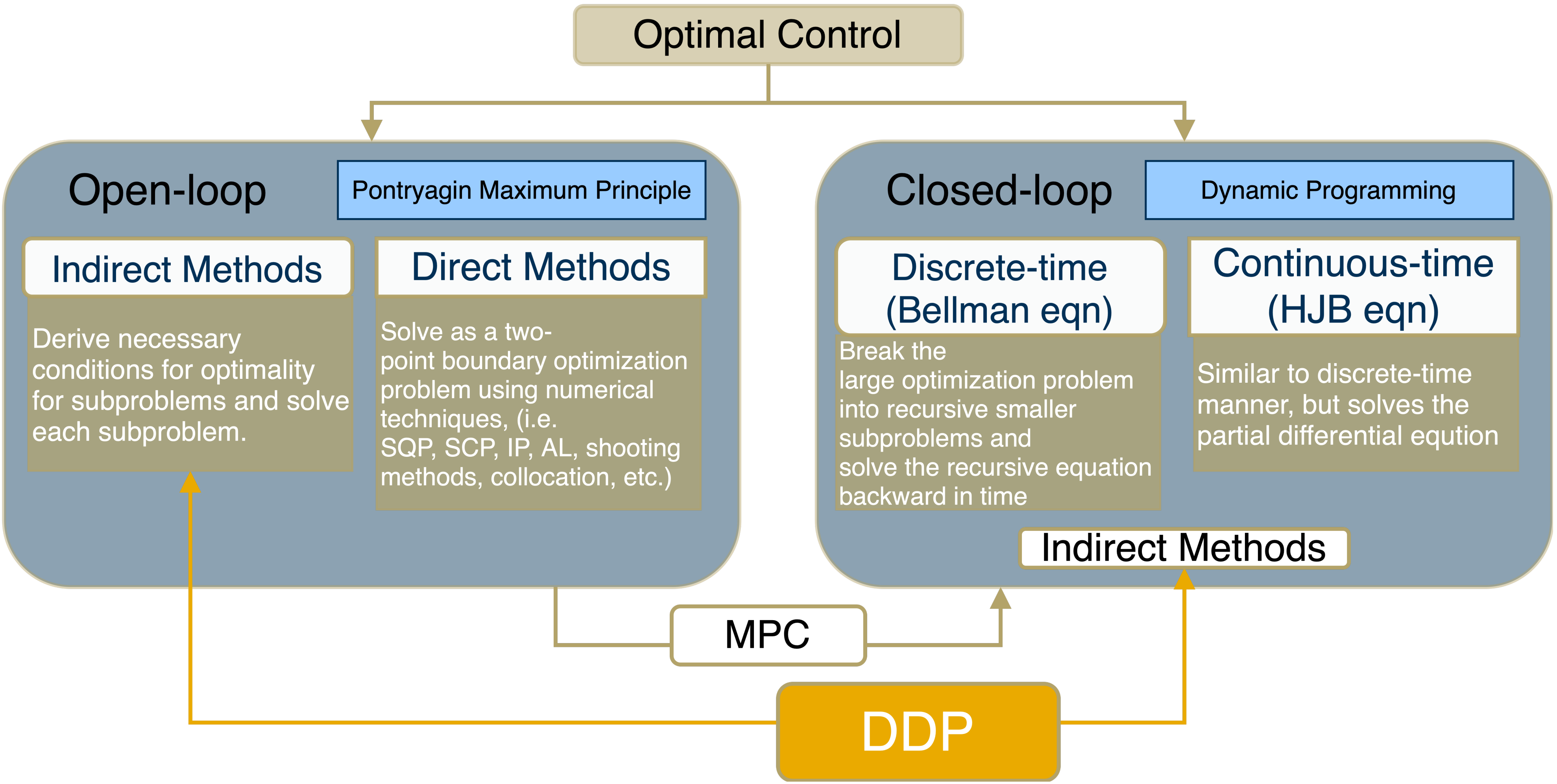
$$\min_{\mathbf{u}} \int_0^T \left[ \|\mathbf{x}(t)\|_p + \|\mathbf{u}(t)\|_p \right] dt$$

$$\text{s.t.} \quad \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t))$$

$$\mathbf{x}(0) = \mathbf{x}_0$$

- Dynamic Programming (Principle of Optimality)
    - compositionality rules of optimal paths
    - Finds a closed-loop solution
    - curse of dimensionality
- Calculus of Variations (Pontryagin Maximum Principle)
    - Extrema of functionals (functional derivative = 0)
    - Finds an open-loop solution

# Optimal Control (2/2)



# Outline

- Motivation
- **Differential Dynamic Programming (DDP)**
- Constrained DDP
- Dynamics and Constraints
- Numerical Simulations
- Future Work and Current Progress
- Conclusion

# Differential Dynamic Programming (DDP)

Consider the finite-horizon discrete-time optimal control problem:

$$V(\mathbf{X}) = \min_{\mathbf{U}} J(\mathbf{X}, \mathbf{U}) = \min_{\mathbf{U}} \left[ \phi\left(\mathbf{x}_N\right) + \sum_{k=0}^{N-1} \ell\left(\mathbf{x}_k, \mathbf{u}_k\right) \right]$$

$$\text{subject to} \quad \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \quad k = 0, \ldots, N-1, \quad \boxed{f, \phi, \ell \in C^2}$$

Using dynamic programming

$$V_k(\mathbf{x}_k) = \min_{\mathbf{u}_k} \left[ \ell(\mathbf{x}_k, \mathbf{u}_k) + V_{k+1}(\mathbf{x}_{k+1}) \right] \quad \text{where } V_N = \phi(\mathbf{x}_N)$$

# Differential Dynamic Programming (DDP)

Consider the finite-horizon discrete-time optimal control problem:

$$V(\mathbf{X}) = \min_{\mathbf{U}} J(\mathbf{X}, \mathbf{U}) = \min_{\mathbf{U}} \left[ \phi\left(\mathbf{x}_N\right) + \sum_{k=0}^{N-1} \ell\left(\mathbf{x}_k, \mathbf{u}_k\right) \right]$$

$$\text{subject to} \quad \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \quad k = 0, \ldots, N-1, \quad \boxed{f, \phi, \ell \in C^2}$$

Using dynamic programming

$$V_k(\mathbf{x}_k) = \min_{\mathbf{u}_k} \left[ \ell(\mathbf{x}_k, \mathbf{u}_k) + V_{k+1}(\mathbf{x}_{k+1}) \right] \quad \text{where } V_N = \phi(\mathbf{x}_N)$$

Quadratically approximate $V_k$ about the nominal trajectory

$$Q(\mathbf{x} + \delta\mathbf{x}, \mathbf{u} + \delta\mathbf{u}) \approx Q(\mathbf{x}, \mathbf{u}) + \begin{bmatrix} Q_{\mathbf{x}} \\ Q_{\mathbf{u}} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \delta\mathbf{x} \\ \delta u \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} Q_{\mathbf{xx}} Q_{\mathbf{xu}} \\ Q_{\mathbf{xu}}^{\mathrm{T}} Q_{\mathbf{uu}} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \end{bmatrix}$$

Minimize $Q$ w.r.t. $\delta\mathbf{u}$, then

$$\delta\mathbf{u}^* = \mathbf{k} + \mathbf{K}\delta\mathbf{x} \text{ with } \mathbf{k} = -Q_{\mathbf{uu}}^{-1} Q_{\mathbf{u}}, \quad \mathbf{K} = -Q_{\mathbf{uu}}^{-1} Q_{\mathbf{ux}}$$

# Differential Dynamic Programming (DDP)

Consider the finite-horizon discrete-time optimal control problem:

$$V(\mathbf{X}) = \min_{\mathbf{U}} J(\mathbf{X}, \mathbf{U}) = \min_{\mathbf{U}} \left[ \phi\left(\mathbf{x}_N\right) + \sum_{k=0}^{N-1} \ell\left(\mathbf{x}_k, \mathbf{u}_k\right) \right]$$

$$\text{subject to} \quad \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \quad k = 0, \ldots, N-1, \quad \boxed{f, \phi, \ell \in C^2}$$

Using dynamic programming

$$V_k(\mathbf{x}_k) = \min_{\mathbf{u}_k} \left[ \ell(\mathbf{x}_k, \mathbf{u}_k) + V_{k+1}(\mathbf{x}_{k+1}) \right] \quad \text{where } V_N = \phi(\mathbf{x}_N)$$

Quadratically approximate $V_k$ about the nominal trajectory

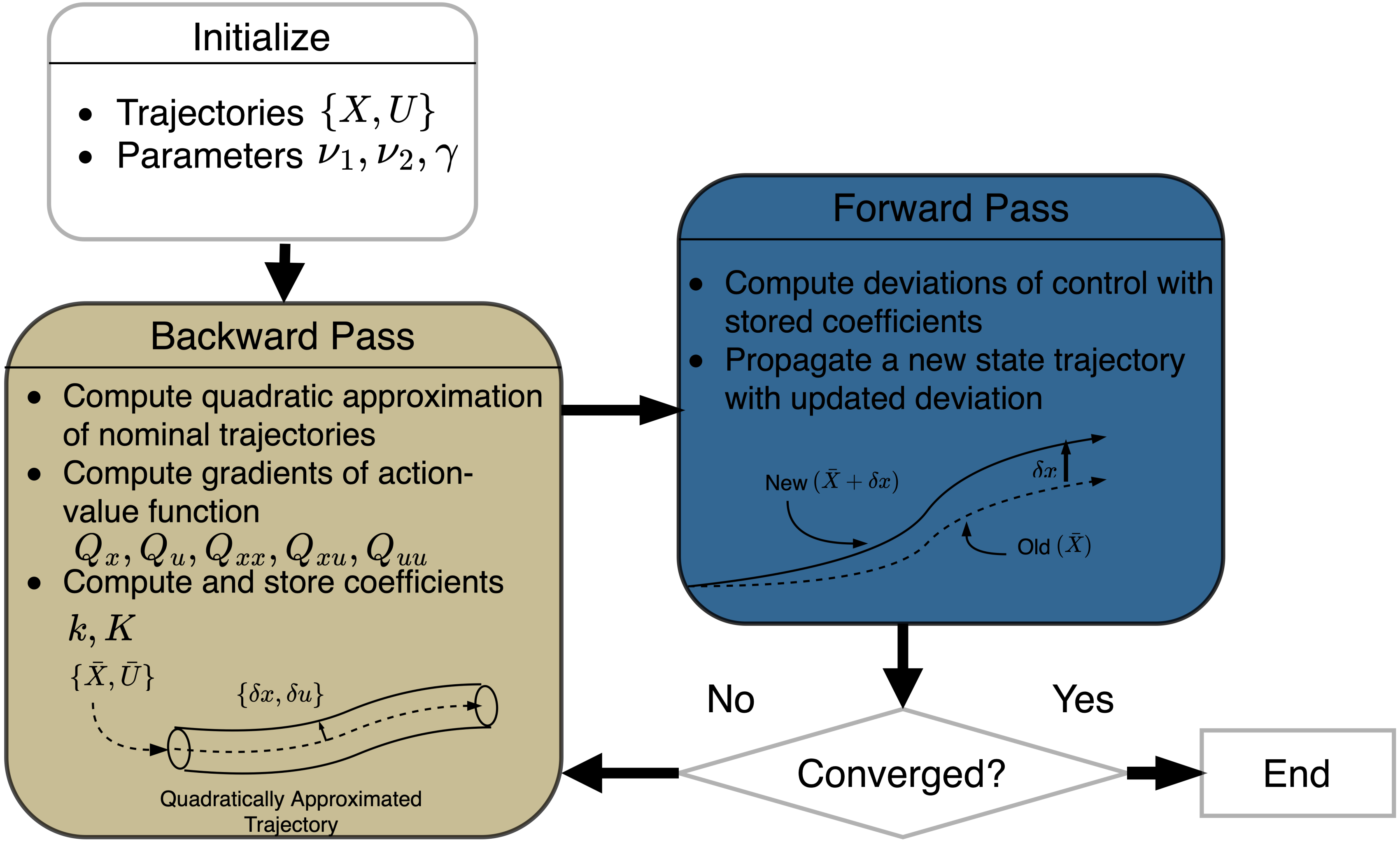
$$Q(\mathbf{x} + \delta\mathbf{x}, \mathbf{u} + \delta\mathbf{u}) \approx Q(\mathbf{x}, \mathbf{u}) + \begin{bmatrix} Q_{\mathbf{x}} \\ Q_{\mathbf{u}} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} Q_{\mathbf{xx}} Q_{\mathbf{xu}} \\ Q_{\mathbf{xu}}^{\mathrm{T}} Q_{\mathbf{uu}} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \end{bmatrix}$$

Minimize $Q$ w.r.t. $\delta\mathbf{u}$, then

$$\delta\mathbf{u}^* = \mathbf{k} + \mathbf{K}\delta\mathbf{x} \ \text{ with } \mathbf{k} = -Q_{\mathbf{uu}}^{-1} Q_{\mathbf{u}}, \quad \mathbf{K} = -Q_{\mathbf{uu}}^{-1} Q_{\mathbf{ux}}$$

Using $\mathbf{u} + \delta\mathbf{u}^*$, propagate trajectory forward in time

# DDP Flow



Initialize
- Trajectories $\{X, U\}$
- Parameters $\nu_1, \nu_2, \gamma$

Backward Pass
- Compute quadratic approximation of nominal trajectories
- Compute gradients of action-value function $Q_x, Q_u, Q_{xx}, Q_{xu}, Q_{uu}$
- Compute and store coefficients $k, K$

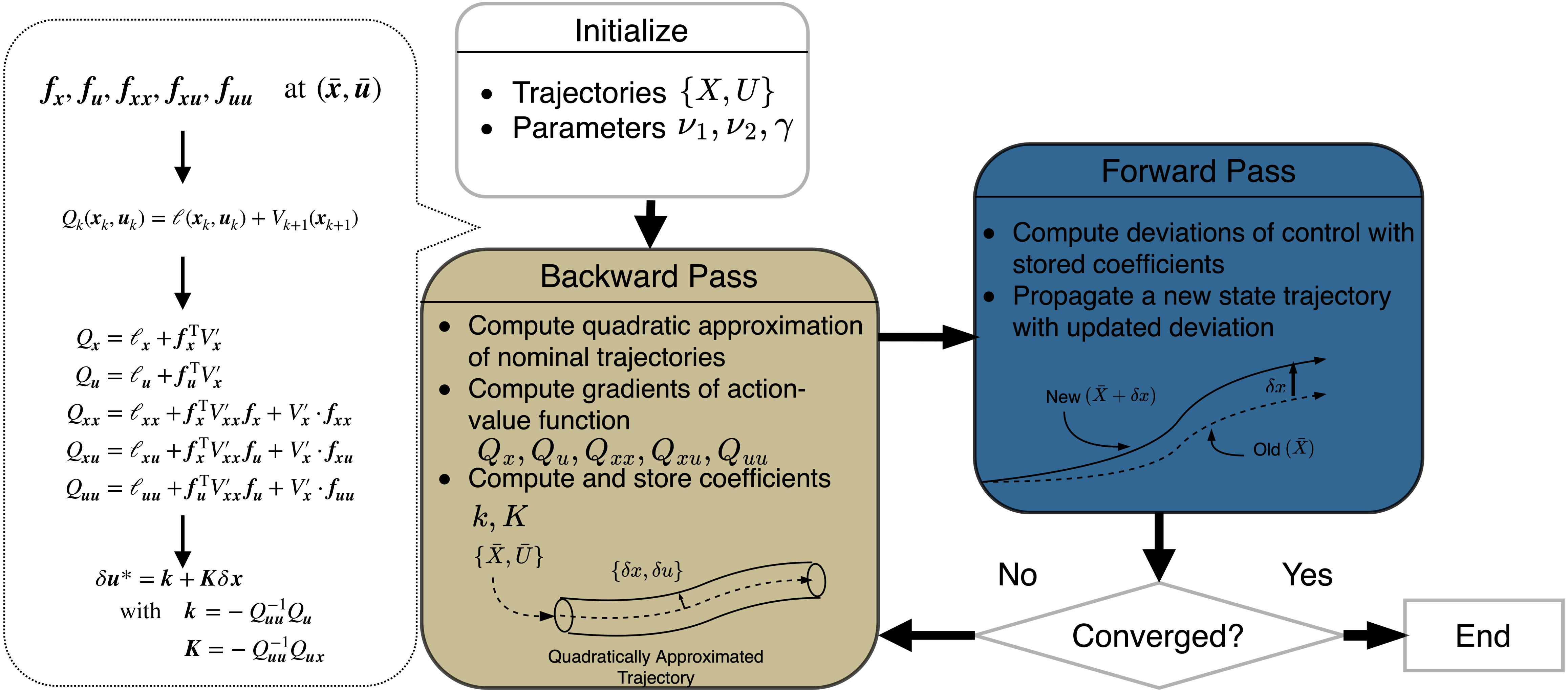
$\{\bar{X}, \bar{U}\}$
$\{\delta x, \delta u\}$
Quadratically Approximated Trajectory

Forward Pass
- Compute deviations of control with stored coefficients
- Propagate a new state trajectory with updated deviation

New $(\bar{X} + \delta x)$
$\delta x$
Old $(\bar{X})$

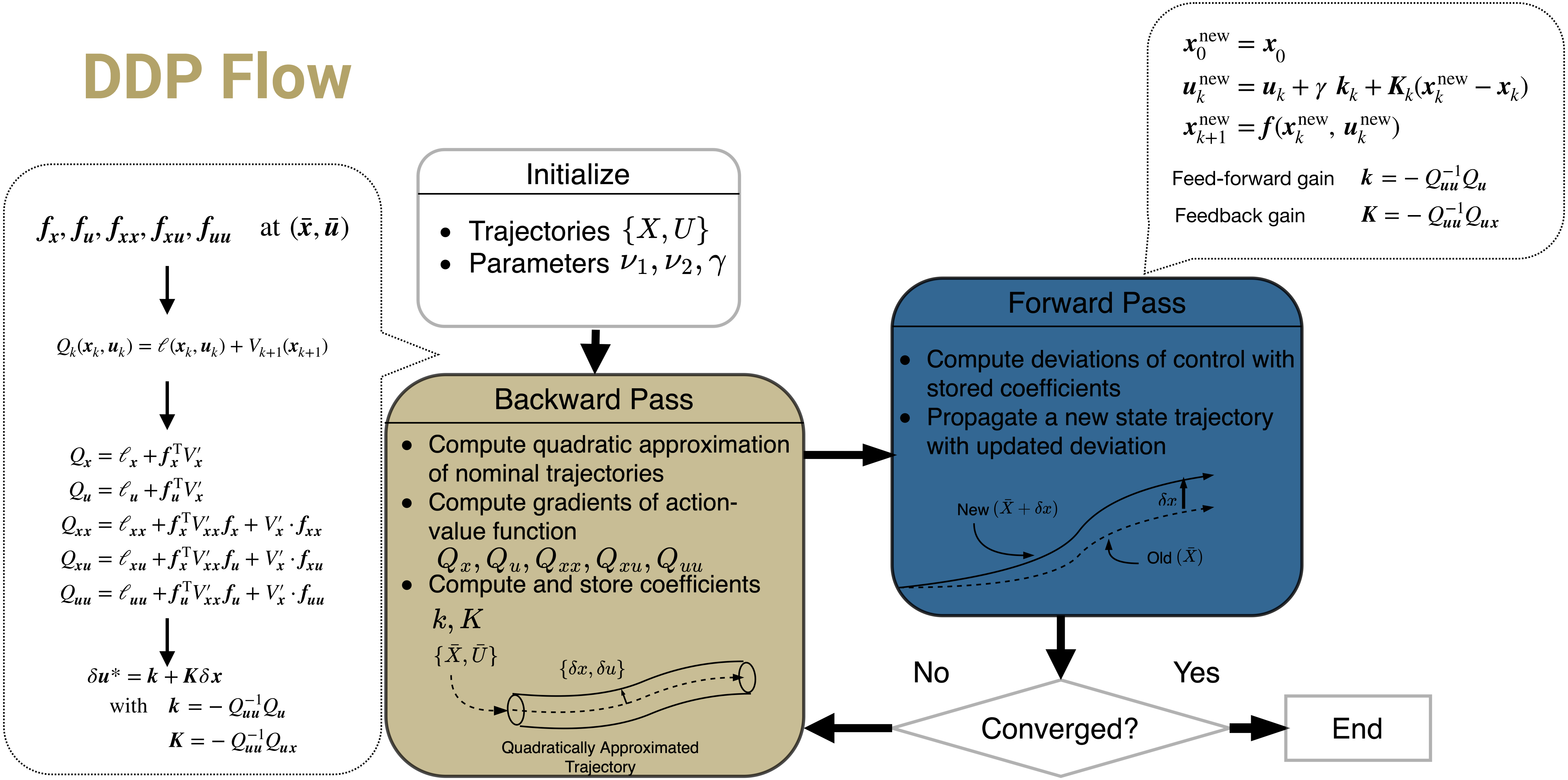
No
Yes
Converged?
End

# DDP Flow



$f_x, f_u, f_{xx}, f_{xu}, f_{uu}$ at $(\bar{x}, \bar{u})$
$Q_k(x_k, u_k) = \ell(x_k, u_k) + V_{k+1}(x_{k+1})$
$Q_x = \ell_x + f_x^T V'_x$
$Q_u = \ell_u + f_u^T V'_x$
$Q_{xx} = \ell_{xx} + f_x^T V'_{xx} f_x + V'_x \cdot f_{xx}$
$Q_{xu} = \ell_{xu} + f_x^T V'_{xx} f_u + V'_x \cdot f_{xu}$
$Q_{uu} = \ell_{uu} + f_u^T V'_{xx} f_u + V'_x \cdot f_{uu}$
$\delta u^* = k + K \delta x$
with $k = -Q_{uu}^{-1} Q_u$
$K = -Q_{uu}^{-1} Q_{ux}$
Initialize
Trajectories $\{X, U\}$
Parameters $\nu_1, \nu_2, \gamma$
Backward Pass
Compute quadratic approximation of nominal trajectories
Compute gradients of action-value function
$Q_x, Q_u, Q_{xx}, Q_{xu}, Q_{uu}$
Compute and store coefficients
$k, K$
$\{\bar{X}, \bar{U}\}$
$\{\delta x, \delta u\}$
Quadratically Approximated Trajectory
Forward Pass
Compute deviations of control with stored coefficients
Propagate a new state trajectory with updated deviation
New $(\bar{X} + \delta x)$
$\delta x$
Old $(\bar{X})$
No
Yes
Converged?
End

# DDP Flow



$f_x, f_u, f_{xx}, f_{xu}, f_{uu}$ at $(\bar{x}, \bar{u})$
$Q_k(x_k, u_k) = \ell(x_k, u_k) + V_{k+1}(x_{k+1})$
$Q_x = \ell_x + f_x^T V'_x$
$Q_u = \ell_u + f_u^T V'_x$
$Q_{xx} = \ell_{xx} + f_x^T V'_{xx} f_x + V'_x \cdot f_{xx}$
$Q_{xu} = \ell_{xu} + f_x^T V'_{xx} f_u + V'_x \cdot f_{xu}$
$Q_{uu} = \ell_{uu} + f_u^T V'_{xx} f_u + V'_x \cdot f_{uu}$
$\delta u^* = k + K\delta x$
with $k = -Q_{uu}^{-1} Q_u$
$K = -Q_{uu}^{-1} Q_{ux}$
Initialize
• Trajectories $\{X, U\}$
• Parameters $\nu_1, \nu_2, \gamma$
Backward Pass
• Compute quadratic approximation of nominal trajectories
• Compute gradients of action-value function
$Q_x, Q_u, Q_{xx}, Q_{xu}, Q_{uu}$
• Compute and store coefficients
$k, K$
$\{\bar{X}, \bar{U}\}$
$\{\delta x, \delta u\}$
Quadratically Approximated Trajectory
Forward Pass
• Compute deviations of control with stored coefficients
• Propagate a new state trajectory with updated deviation
New $(\bar{X} + \delta x)$
$\delta x$
Old $(\bar{X})$
$x_0^{new} = x_0$
$u_k^{new} = u_k + \gamma\, k_k + K_k(x_k^{new} - x_k)$
$x_{k+1}^{new} = f(x_k^{new}, u_k^{new})$
Feed-forward gain $k = -Q_{uu}^{-1} Q_u$
Feedback gain $K = -Q_{uu}^{-1} Q_{ux}$
No
Yes
Converged?
End

# Outline

- Motivation
- Differential Dynamic Programming (DDP)
- Constrained DDP
- Dynamics and Constraints
- Numerical Simulations
- Future Work and Current Progress
- Conclusion

# Constrained DDP

- CDDP can deal with state and control constraints using the primal-dual interior point method (feasible and infeasible)
- Optimality of control is satisfied by the perturbed KKT system

# Constrained DDP

- CDDP can deal with state and control constraints using the primal-dual interior point method (feasible and infeasible)
- Optimality of control is satisfied by the perturbed KKT system

$$
\begin{aligned}
\nabla_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}, \boldsymbol{s}) &= 0 \\
\nabla_{\boldsymbol{u}} L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}, \boldsymbol{s}) &= 0 \\
\nabla_{\boldsymbol{s}} L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}, \boldsymbol{s}) &= 0 \\
\operatorname{diag}(\boldsymbol{\lambda}) \boldsymbol{c}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\mu} &= 0 \\
\boldsymbol{c}(\boldsymbol{x}, \boldsymbol{u}) \leq 0, \quad \boldsymbol{\lambda} &\geq 0
\end{aligned}
$$

# CDDP

Consider the finite-horizon discrete-time optimal control problem:

$$\min_{\mathbf{U}} J(\mathbf{X}, \mathbf{U}) = \min_{\mathbf{U}} \left[ \phi\left(\mathbf{x}_N\right) + \sum_{k=0}^{N-1} \ell\left(\mathbf{x}_k, \mathbf{u}_k\right) \right]$$

$$\text{subject to} \quad \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k)$$

$$c(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \quad k = 0, \ldots, N-1, \quad f, c, \ell, \phi \in C^2$$

Using dynamic programming

$$V_k\left(\mathbf{x}_k\right) = \min_{\mathbf{u}_k \text{ s.t. } \mathbf{c}(\mathbf{x}, \mathbf{u}) \leq 0} \left[ \ell\left(\mathbf{x}_k, \mathbf{u}_k\right) + V_{k+1}\left(\mathbf{x}_{k+1}\right) \right]$$

# CDDP

Consider the finite-horizon discrete-time optimal control problem:

$$\min_{\mathbf{U}} J(\mathbf{X},\mathbf{U}) = \min_{\mathbf{U}} \left[ \phi\left(\mathbf{x}_N\right) + \sum_{k=0}^{N-1} \ell\left(\mathbf{x}_k, \mathbf{u}_k\right) \right]$$

$$\text{subject to} \quad \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k)$$

$$c(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \quad k = 0, \ldots, N-1, \quad f, c, \ell, \phi \in C^2$$

Using dynamic programming

$$V_k\left(\mathbf{x}_k\right) = \min_{\mathbf{u}_k \text{ s.t. } \mathbf{c}(\mathbf{x},\mathbf{u})\leq 0} \left[ \ell\left(\mathbf{x}_k, \mathbf{u}_k\right) + V_{k+1}\left(\mathbf{x}_{k+1}\right) \right]$$

Applying the minimax DDP technique, this recursive equation can be transformed into,

$$V_k\left(\mathbf{x}_k\right) = \min_{\mathbf{u}_k} \max_{\lambda_k \geq 0} \left[ \mathscr{L}\left(\mathbf{x}_k, \boldsymbol{u}_k, \boldsymbol{\lambda}_k\right) + V_{k+1}\left(\mathbf{x}_{k+1}\right) \right] \quad \text{where} \quad \mathscr{L}(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\lambda}_k) = \ell(\mathbf{x}_k, \mathbf{u}_k) + \boldsymbol{\lambda}_k^{\mathrm{T}} \boldsymbol{c}(\mathbf{x}_k, \mathbf{u}_k)$$

# CDDP

Consider the finite-horizon discrete-time optimal control problem:

$$\min_{\mathbf{U}} J(\mathbf{X},\mathbf{U}) = \min_{\mathbf{U}} \left[ \phi\left(\mathbf{x}_N\right) + \sum_{k=0}^{N-1} \ell\left(\mathbf{x}_k, \mathbf{u}_k\right) \right]$$

$$\text{subject to} \quad \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k)$$

$$c(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \quad k = 0, \ldots, N-1, \quad f, c, \ell, \phi \in C^2$$

Using dynamic programming

$$V_k\left(\mathbf{x}_k\right) = \min_{\mathbf{u}_k \text{ s.t. } \mathbf{c}(\mathbf{x},\mathbf{u}) \leq 0} \left[ \ell\left(\mathbf{x}_k, \mathbf{u}_k\right) + V_{k+1}\left(\mathbf{x}_{k+1}\right) \right]$$

Applying the minimax DDP technique, this recursive equation can be transformed into,

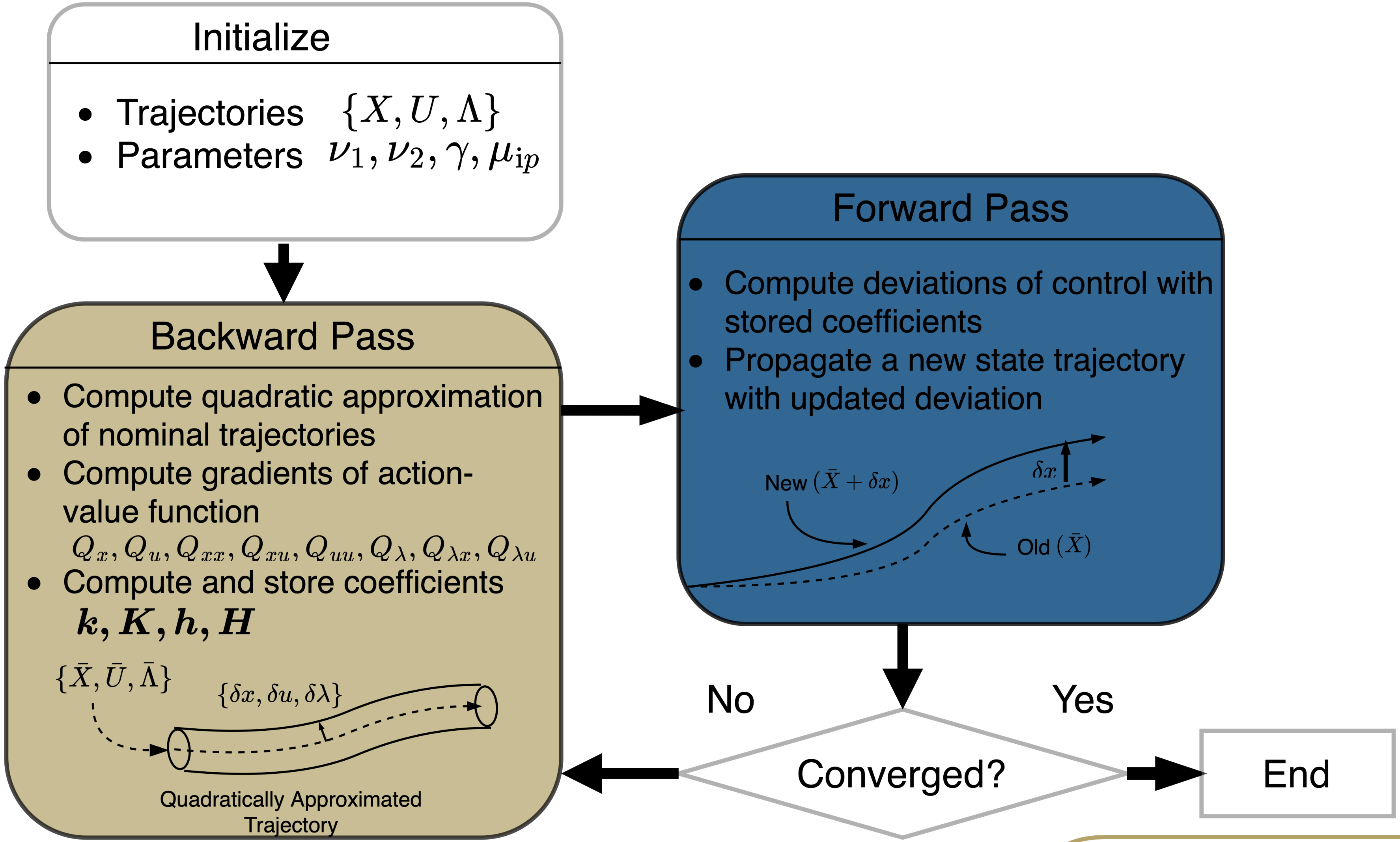
$$V_k\left(\mathbf{x}_k\right) = \min_{\mathbf{u}_k} \max_{\lambda_k \geq 0} \left[ \mathscr{L}\left(\mathbf{x}_k, \boldsymbol{u}_k, \boldsymbol{\lambda}_k\right) + V_{k+1}\left(\mathbf{x}_{k+1}\right) \right] \quad \text{where} \quad \mathscr{L}(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\lambda}_k) = \ell(\mathbf{x}_k, \mathbf{u}_k) + \boldsymbol{\lambda}_k^{\mathrm{T}} \boldsymbol{c}(\mathbf{x}_k, \mathbf{u}_k)$$

Now, quadratically approximate $V_k$ about the nominal trajectory and minimize $Q$ w.r.t. $\delta\mathbf{u}$ to find

$$\delta\mathbf{u}^* = \mathbf{k} + \mathbf{K}\delta\mathbf{x} \text{ and } \delta\lambda^* = \mathbf{h} + \mathbf{H}\delta\mathbf{x}$$

Using $\mathbf{u} + \delta\mathbf{u}^*$ and $\lambda + \delta\lambda^*$, propagate trajectory forward in time

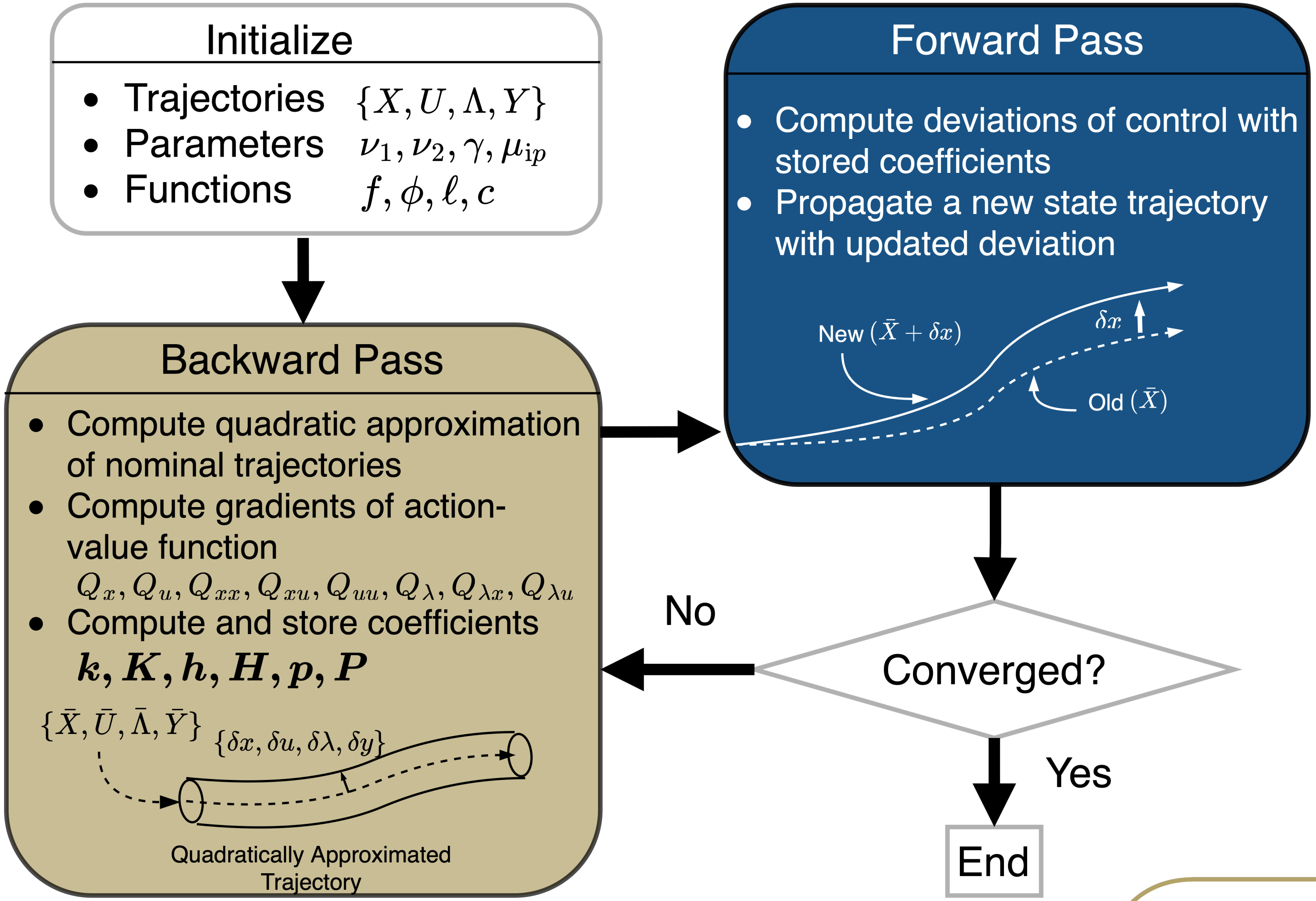
# CDDP Flow (feasible)






Daniel Guggenheim School of Aerospace Engineering
SSDL | Space Systems Design Laboratory

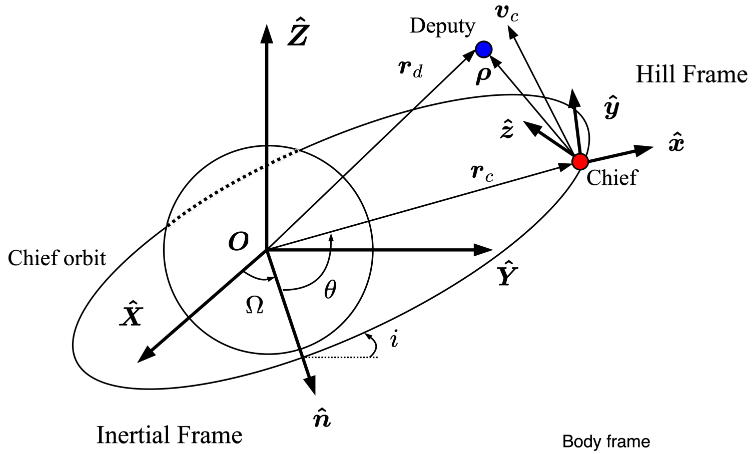
# DDP Flow (Infeasible)



# Outline

- Motivation
- Differential Dynamic Programming (DDP)
- Constrained DDP
- Dynamics and Constraints
- Numerical Simulations
- Future Work and Current Progress
- Conclusion

# Dynamics (6DoF) and Constraints



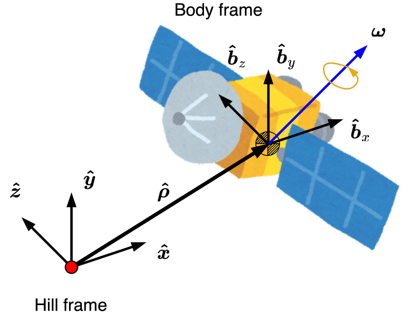

- Dynamics
  - Clohessy-Wiltshire linear model
  - quaternion attitude model

$$\mathbf{x} = \begin{bmatrix} \mathbf{r} \\ \mathbf{v} \\ \mathbf{q} \\ \boldsymbol{\omega} \end{bmatrix} \in \mathbb{R}^{13}, \quad \dot{\mathbf{x}} = \begin{bmatrix} \mathbf{v} \\ \frac{1}{m} f^{HCW}(\mathbf{r}, \mathbf{v}, \mathbf{u}) \\ \frac{1}{2}\mathbf{q} \odot \boldsymbol{\omega} \\ J^{-1}\left(\tau - \boldsymbol{\omega} \times (J\boldsymbol{\omega})\right) \end{bmatrix}$$

- Performance Index
  - quadratic cost

$$\phi(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{R}_1\mathbf{x}, \quad \ell(\mathbf{u}) = \mathbf{u}^{\mathrm{T}}\mathbf{R}_2\mathbf{u}$$

- Constraints




$$t_T \text{ fixed}$$
$$\mathbf{x}(t_0) = \mathbf{x}_0$$
$$\mathbf{x}(t_T) = \mathbf{x}_T$$
$$-\mathbf{u}_{min} \leq \mathbf{u}(t) \leq \mathbf{u}_{max}$$
$$39 \text{ m} \leq \|\mathbf{r}_{1:2}(t)\|_2$$

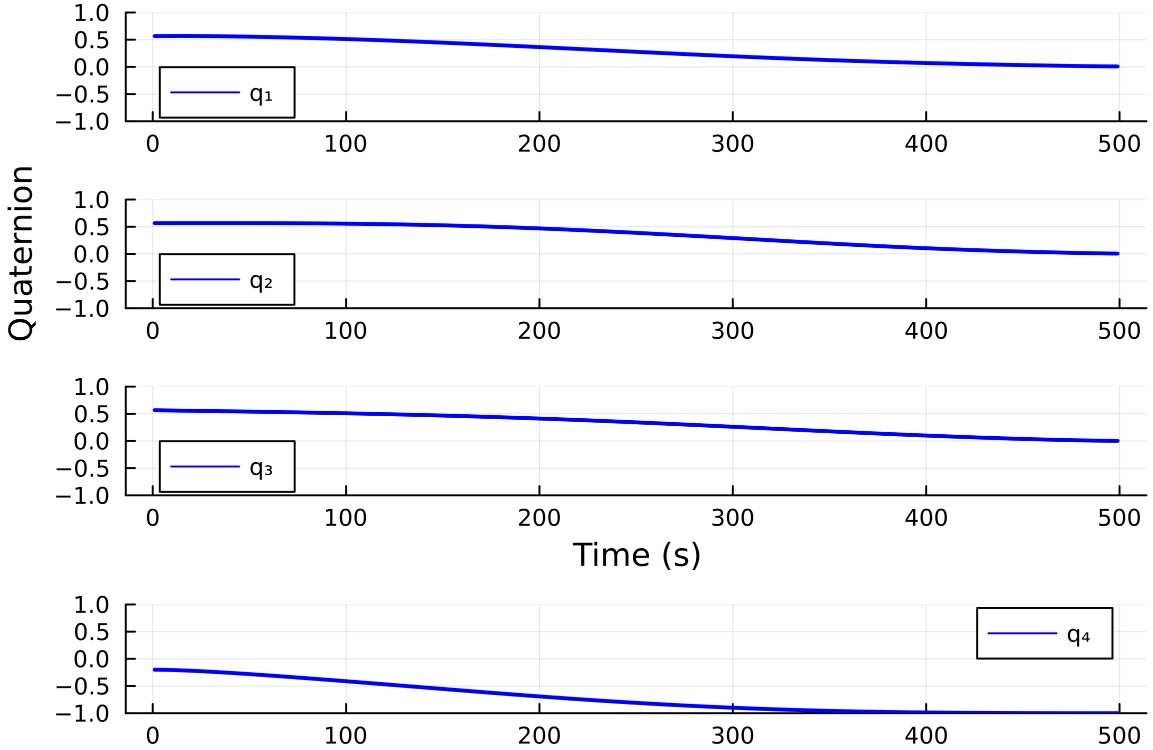
in-plane safe constraint

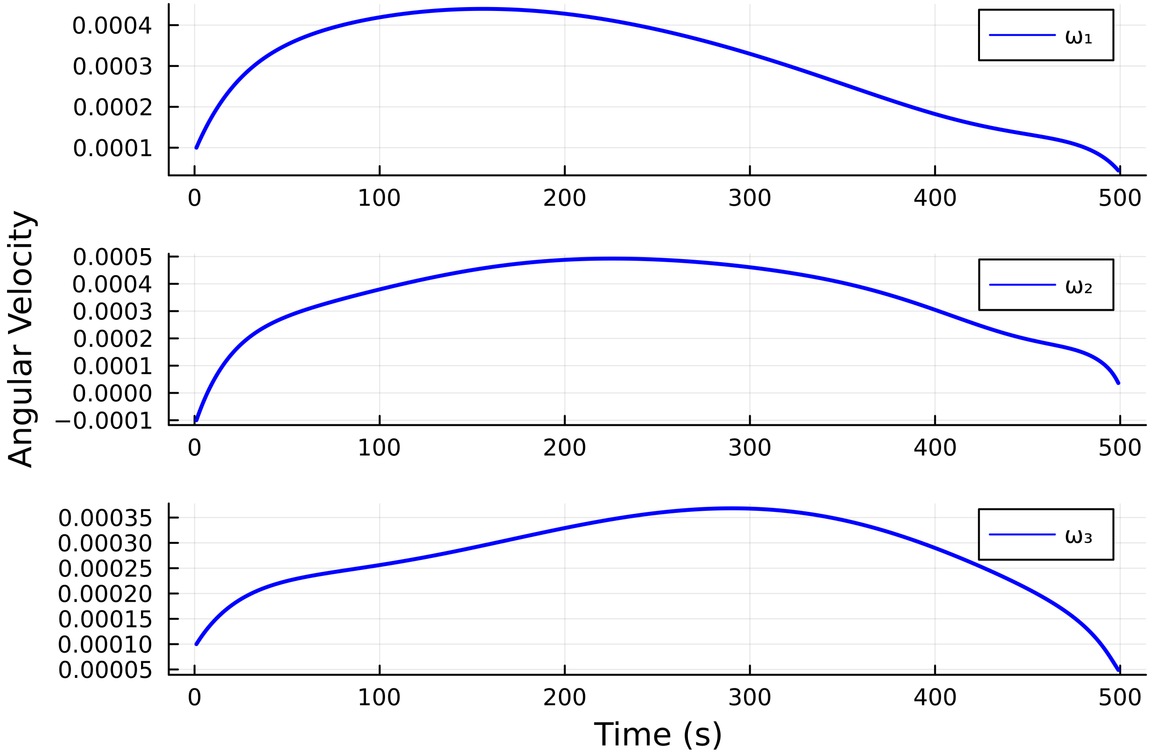
# Outline

- Motivation
- Differential Dynamic Programming (DDP)
- Constrained DDP
- Dynamics and Constraints
- Numerical Simulations
- Future Work and Current Progress
- Conclusion

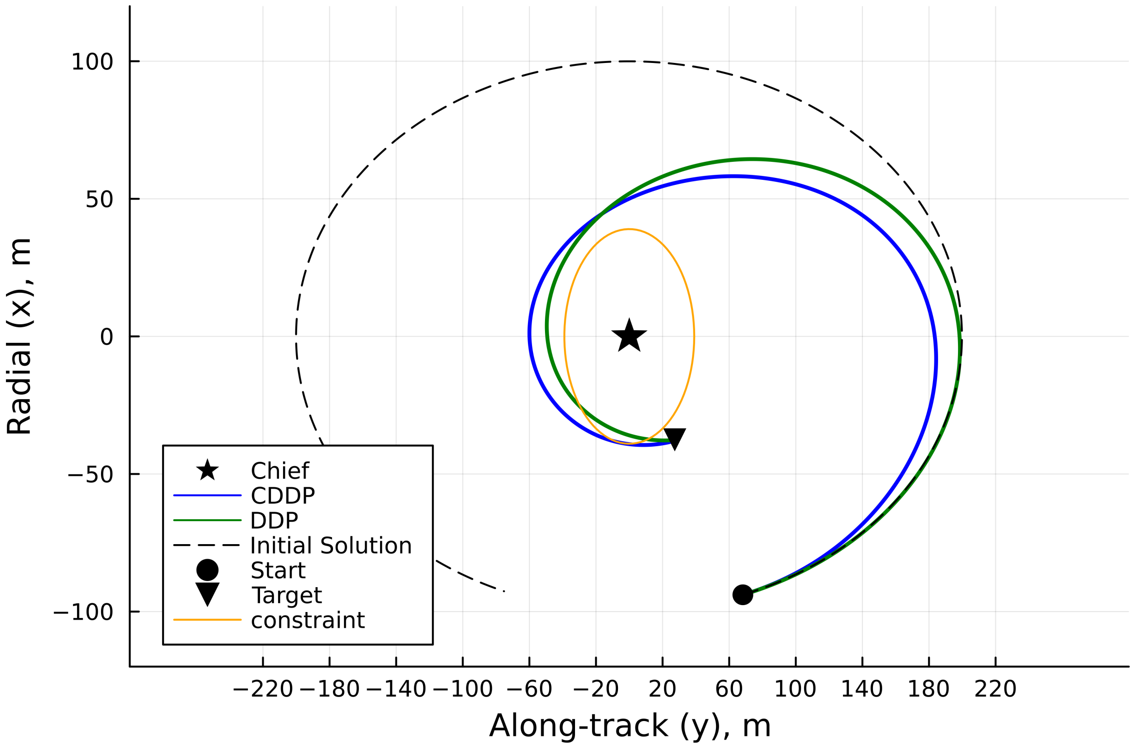
# Initially Feasible Case: Attitude

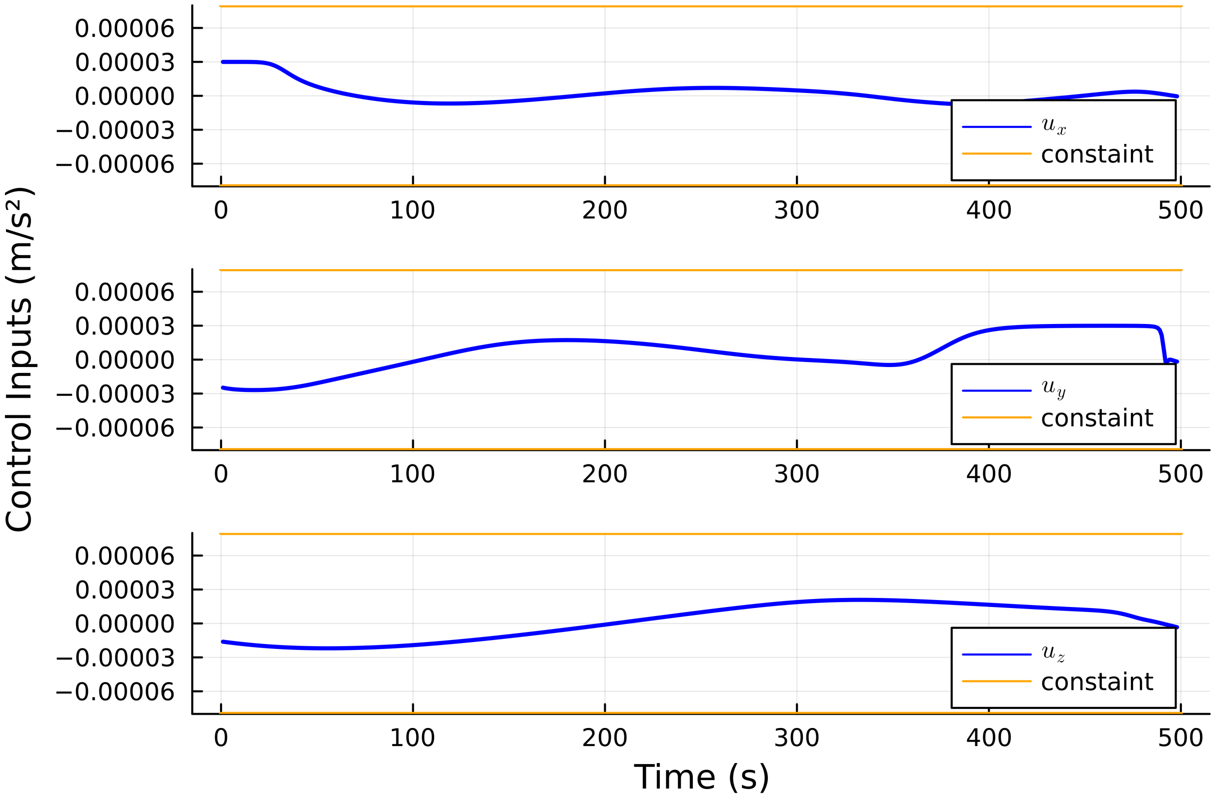
- $\boldsymbol{q}(t_0) = [0.5657, 0.5657, 0.5657, -0.2]^T \rightarrow \boldsymbol{q}(T) = [0,0,0,-1]^T$
- $\boldsymbol{\omega}(t_0) = [0,0,0]^T \rightarrow \boldsymbol{\omega}(T) = [0,0,0]^T$



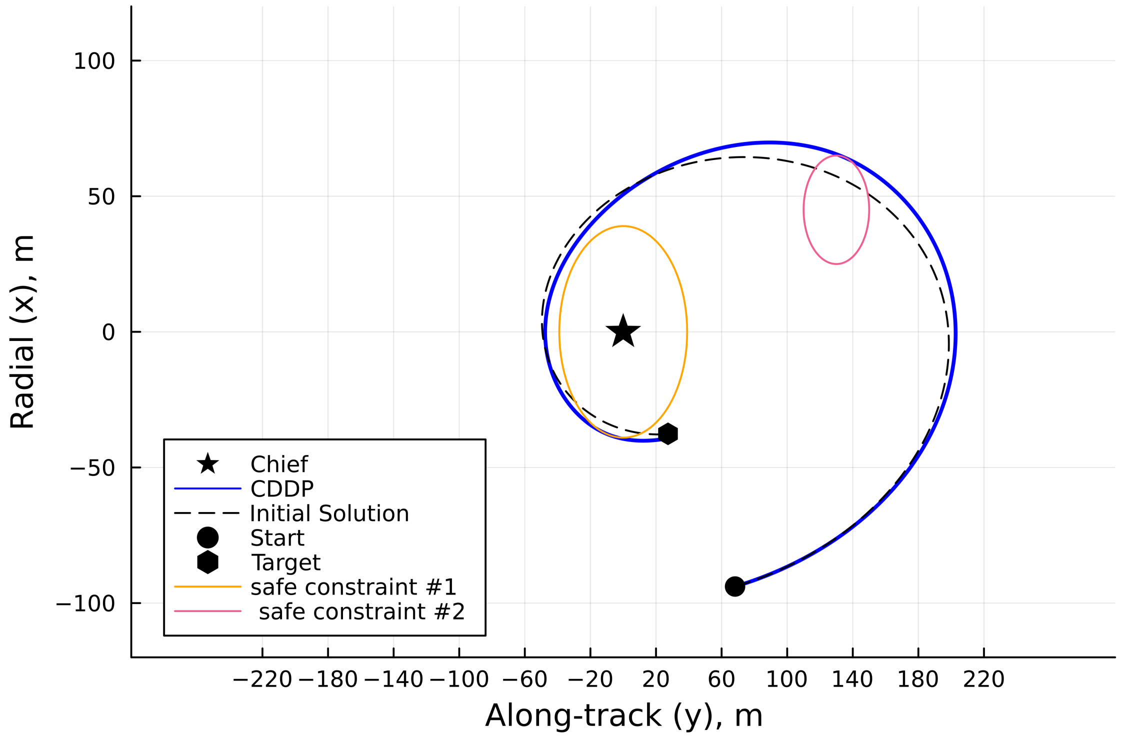



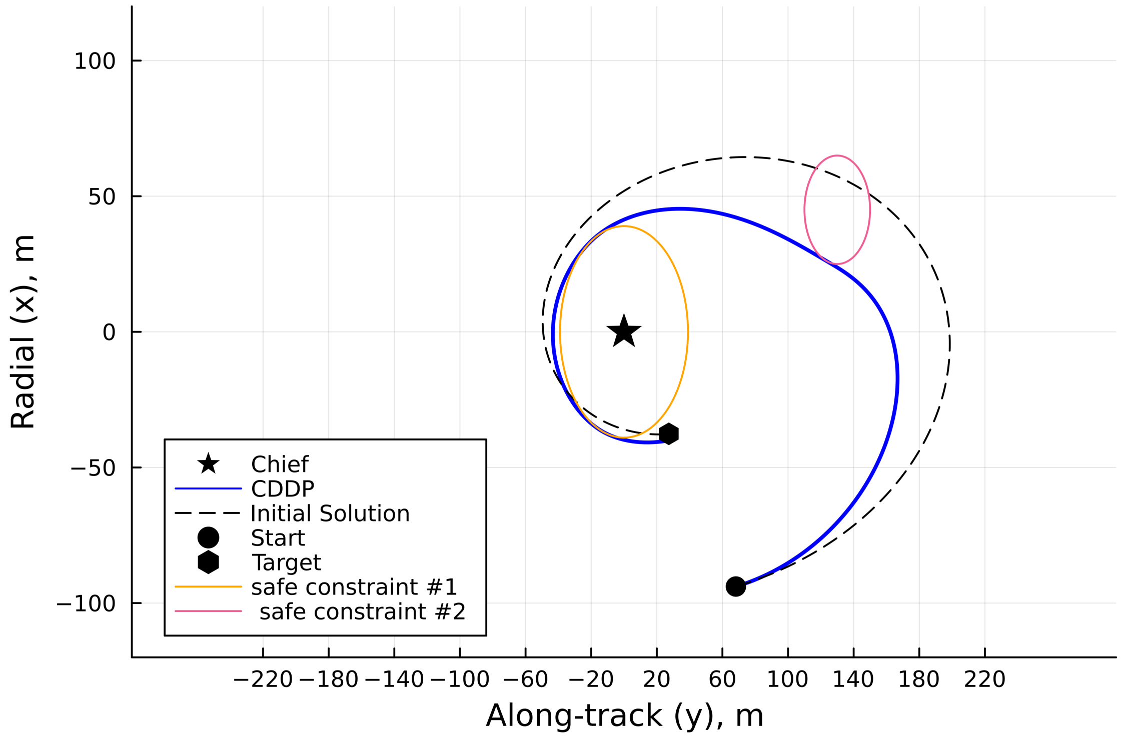
# Initially Feasible Case: Path



# Initially Infeasible Case: Path



# Outline

- Motivation
- Differential Dynamic Programming (DDP)
- Constrained DDP
- Dynamics and Constraints
- Numerical Simulations
- **Future Work and Current Progress**
- Conclusion

# Further Motivation and Current Progress

- CDDP under uncertainty
  - dynamics and constraints are no longer deterministic
  - can't use deterministic CDDP directly
- reachability of constrained optimal control

# Chance-constrained Differential Dynamic Programming

$$\min_{\mathbf{U}} J(\mathbf{X}, \mathbf{U}) = \min_{\mathbf{U}} \mathbb{E}\left[\phi(\mathbf{x}_N) + \sum_{k=0}^{N-1} \ell(\mathbf{x}_k, \mathbf{u}_k)\right]$$

$$\begin{aligned} \text{subject to} \quad & \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + G(\mathbf{x}_k, \mathbf{u}_k)\mathbf{w}_k \\ & \Pr[c(\mathbf{x}_k, \mathbf{u}_k) \leq 0] > 1 - \varepsilon \\ & k = 0, \ldots, N-1, \quad f, c, \phi, \ell \in C^2 \end{aligned}$$

# Conclusion

- DDP and CDDP showed successful fast convergence when it’s applied to the linear relative motion model and quaternion model
- The academic and practical values of chance-constrained DDP will further be discussed

Thank you for your attention!

I appreciate your comments and questions!

Good Luck with your Projects and Exams!

Tomohiro Sasaki
tomohiro.sasaki@gatech.edu