Problem definition

Karel Zimmermann

Prerequisites

- **Mathematical analysis (B0B01MA2):** gradient, Jacobian, Hessian, multidimensional Taylor polynomial
- **Optimization (B0B33OPT):** Gauss-Newton method, Levenberg Marquardt method, full Newton method
- **Linear algebra (B0B01LAG):** pseudo-inverse, SVD decomposition, least-squares method
- **Probability theory (B0B01PST):** multivariate gaussian probability, Bayes theorem
- **Statistics (B0B01PST):** maximum likelihood and maximum aposteriori estimate
- **Programming (B3B33ALP + B3B36PRG):** python + linux

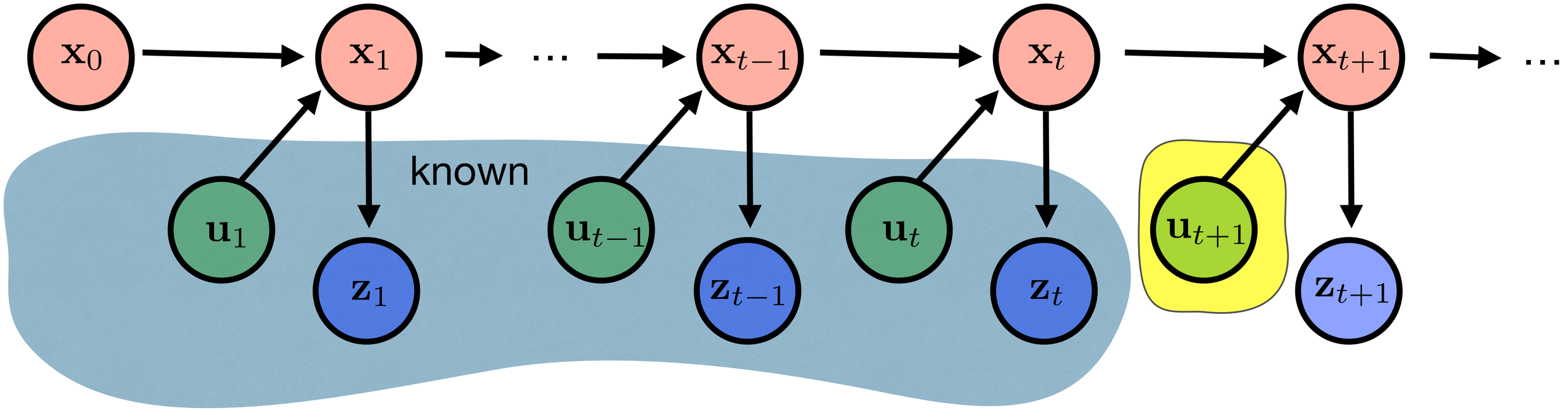
Problem definition

States: $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t \in \mathcal{R}^n$

Actions: $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathcal{R}^m$

Measurements: $\mathbf{z}_1, \ldots, \mathbf{z}_t \in \mathcal{R}^k$

Algorithm: $\mathbf{u}_{t+1} = \pi(\mathbf{z}_{1:t}, \mathbf{u}_{1:t})$



Problem definition

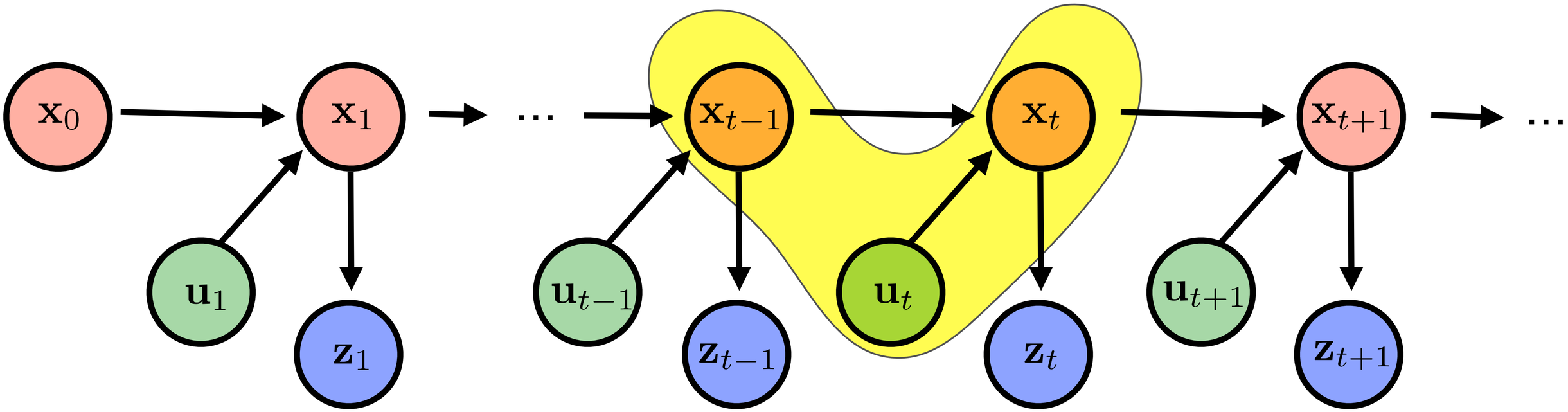
States: $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t \in \mathcal{R}^n$

Actions: $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathcal{R}^m$

Measurements: $\mathbf{z}_1, \ldots, \mathbf{z}_t \in \mathcal{R}^k$

Algorithm: $\mathbf{u}_{t+1} = \pi(\mathbf{z}_{1:t}, \mathbf{u}_{1:t})$

Rewards: $r_t = r(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{x}_t) \in \mathcal{R}$



Problem definition

States: $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t \in \mathcal{R}^n$

Actions: $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathcal{R}^m$

Measurements: $\mathbf{z}_1, \ldots, \mathbf{z}_t \in \mathcal{R}^k$

Algorithm: $\mathbf{u}_{t+1} = \pi(\mathbf{z}_{1:t}, \mathbf{u}_{1:t})$

Rewards: $r_t = r(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{x}_t) \in \mathcal{R}$

Problem definition

States: $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t \in \mathcal{R}^n$

Actions: $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathcal{R}^m$

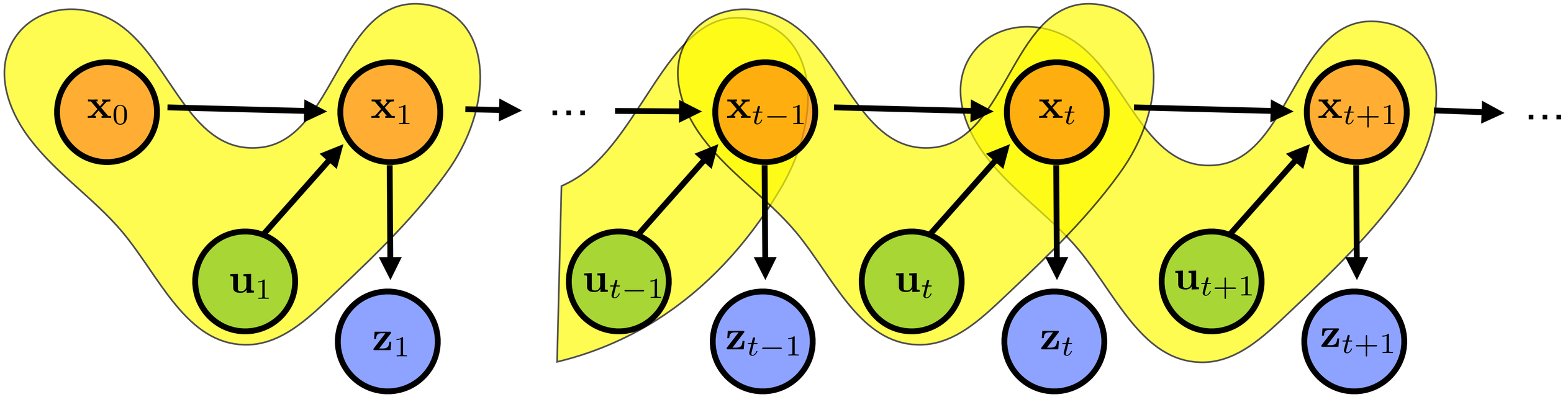
Measurements: $\mathbf{z}_1, \ldots, \mathbf{z}_t \in \mathcal{R}^k$

Algorithm: $\mathbf{u}_{t+1} = \pi(\mathbf{z}_{1:t}, \mathbf{u}_{1:t})$

Rewards: $r_t = r(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{x}_t) \in \mathcal{R}$

Criterion: $J_\pi = \mathbb{E}_{\tau \sim \pi}\{ \sum_{r_t \sim \tau} \gamma^t r_t \} \in \mathcal{R}$

Goal: $\pi^* = \arg\max_\pi J_\pi$



Problem definition

Complete states: $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t \in \mathcal{R}^n$

Actions: $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathcal{R}^m$

Measurements: $\mathbf{z}_1, \ldots, \mathbf{z}_t \in \mathcal{R}^k$

Algorithm: $\mathbf{u}_{t+1} = \pi(\mathbf{z}_{1:t}, \mathbf{u}_{1:t})$

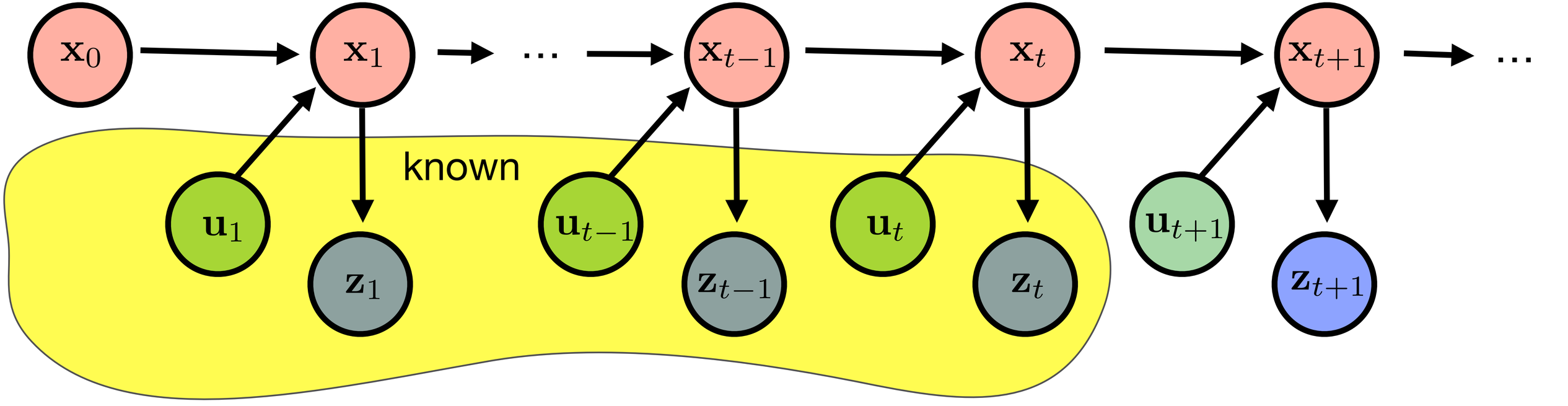
Rewards: $r_t = r(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{x}_t) \in \mathcal{R}$

Criterion: $J_\pi = \mathbb{E}_{\tau \sim \pi}\{ \sum_{r_t \sim \tau} \gamma^t r_t \} \in \mathcal{R}$

Goal: $\pi^* = \arg\max_\pi J_\pi$

Algorithm: $\mathbf{z}_0, \mathbf{u}_1, \mathbf{z}_1, \ldots$ => estimate $p(\mathbf{x}_t | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$ => $\pi(\mathbf{x}_t)$ decide following action $\mathbf{u}_{t+1}$

perception (local, SLAM, object detection) control (planning, RL, opt.control



Problem definition

States: $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t \in \mathcal{R}^n$

Actions: $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathcal{R}^m$

Measurements: $\mathbf{z}_1, \ldots, \mathbf{z}_t \in \mathcal{R}^k$

Algorithm: $\mathbf{u}_{t+1} = \pi(\mathbf{z}_{1:t}, \mathbf{u}_{1:t})$

Rewards: $r_t = r(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{x}_t) \in \mathcal{R}$

Criterion: $J_\pi = \mathbb{E}_{\tau \sim \pi}\{ \sum_{r_t \sim \tau} \gamma^t r_t \} \in \mathcal{R}$

Goal: $\pi^* = \arg\max_\pi J_\pi$

Examples:

Problem definition

States: $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t \in \mathcal{R}^n$

Actions: $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathcal{R}^m$

Measurements: $\mathbf{z}_1, \ldots, \mathbf{z}_t \in \mathcal{R}^k$

Algorithm: $\mathbf{u}_{t+1} = \pi(\mathbf{z}_{1:t}, \mathbf{u}_{1:t})$

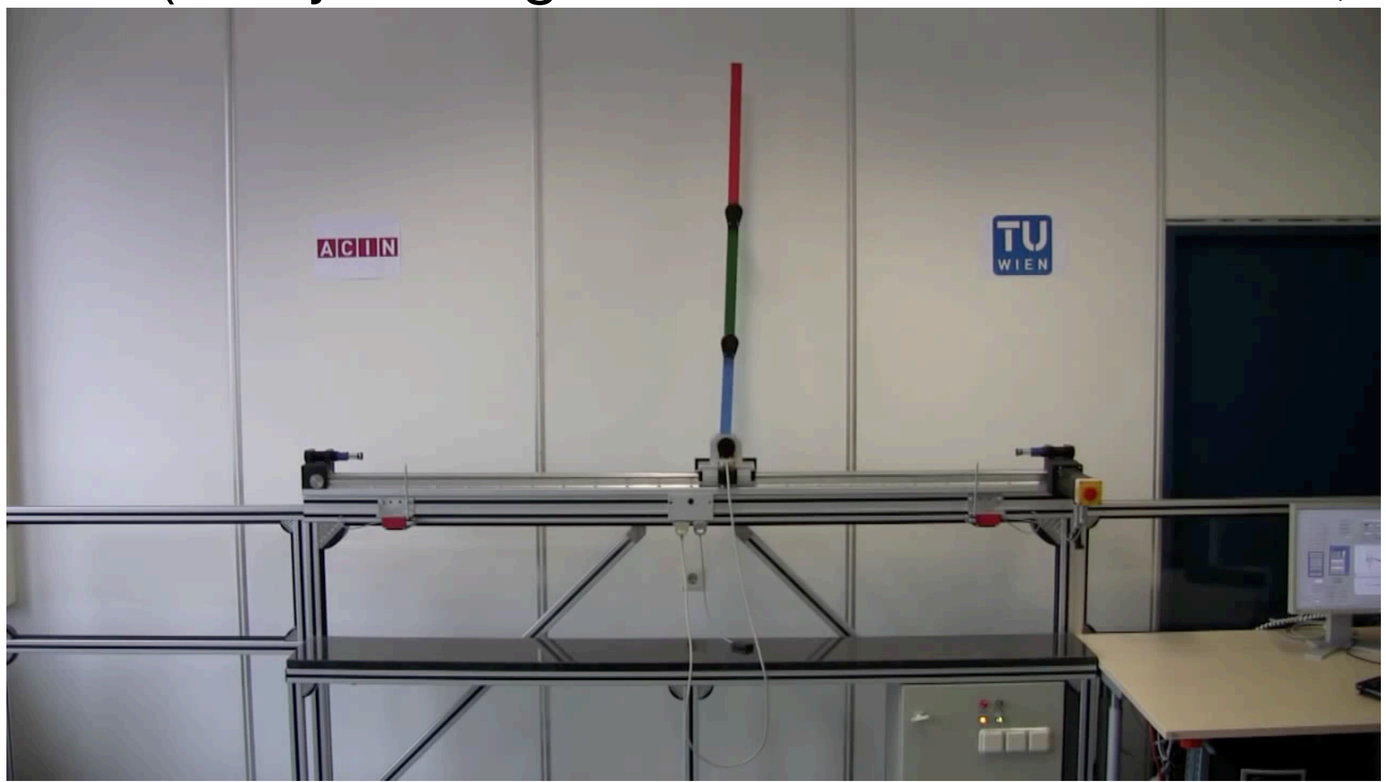
Rewards: $r_t = r(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{x}_t) \in \mathcal{R}$

Criterion: $J_\pi = \mathbb{E}_{\tau \sim \pi}\{ \sum_{r_t \sim \tau} \gamma^t r_t \} \in \mathcal{R}$

Goal: $\pi^* = \arg\max_\pi J_\pi$

- Balancing pendulum (z=x: joint angles/velocities/accelerations, u: base vel., r: height)

Examples:



Problem definition

States: $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t \in \mathcal{R}^n$

Actions: $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathcal{R}^m$

Measurements: $\mathbf{z}_1, \ldots, \mathbf{z}_t \in \mathcal{R}^k$

Algorithm: $\mathbf{u}_{t+1} = \pi(\mathbf{z}_{1:t}, \mathbf{u}_{1:t})$

Rewards: $r_t = r(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{x}_t) \in \mathcal{R}$

Criterion: $J_\pi = \mathbb{E}_{\tau \sim \pi}\{ \sum_{r_t \sim \tau} \gamma^t r_t \} \in \mathcal{R}$

Goal: $\pi^* = \arg\max_\pi J_\pi$

Examples:

- Balancing pendulum (z=x: joint angles/velocities/accelerations, u: base vel., r: height)

Problem definition

States: $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t \in \mathcal{R}^n$

Actions: $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathcal{R}^m$

Measurements: $\mathbf{z}_1, \ldots, \mathbf{z}_t \in \mathcal{R}^k$

Algorithm: $\mathbf{u}_{t+1} = \pi(\mathbf{z}_{1:t}, \mathbf{u}_{1:t})$

Rewards: $r_t = r(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{x}_t) \in \mathcal{R}$

Criterion: $J_\pi = \mathbb{E}_{\tau \sim \pi}\{ \sum_{r_t \sim \tau} \gamma^t r_t \} \in \mathcal{R}$

Goal: $\pi^* = \arg\max_\pi J_\pi$

- RC helicopter tricks (z: body pose+angles/vel/acc, u: motor torques, r: ???)

Examples:

Problem definition

States: $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t \in \mathcal{R}^n$

Actions: $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathcal{R}^m$

Measurements: $\mathbf{z}_1, \ldots, \mathbf{z}_t \in \mathcal{R}^k$

Algorithm: $\mathbf{u}_{t+1} = \pi(\mathbf{z}_{1:t}, \mathbf{u}_{1:t})$

Rewards: $r_t = r(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{x}_t) \in \mathcal{R}$

Criterion: $J_\pi = \mathbb{E}_{\tau \sim \pi}\{ \sum_{r_t \sim \tau} \gamma^t r_t \} \in \mathcal{R}$

Goal: $\pi^* = \arg\max_\pi J_\pi$

Examples:

- Balancing pendulum (z=x: joint angles/velocities/accelerations, u: base vel., r: height)
- RC helicopter tricks (z: body pose+angles/vel/acc, u: motor torques, r: ???)

Problem definition

States: $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t \in \mathcal{R}^n$

Actions: $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathcal{R}^m$

Measurements: $\mathbf{z}_1, \ldots, \mathbf{z}_t \in \mathcal{R}^k$

Algorithm: $\mathbf{u}_{t+1} = \pi(\mathbf{z}_{1:t}, \mathbf{u}_{1:t})$

Rewards: $r_t = r(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{x}_t) \in \mathcal{R}$

Criterion: $J_\pi = \mathbb{E}_{\tau \sim \pi}\{ \sum_{r_t \sim \tau} \gamma^t r_t \} \in \mathcal{R}$

Goal: $\pi^* = \arg\max_\pi J_\pi$

- Agent playing Atari games (z: RGB, u: joystick, r: score/lives, x: ???)

Examples:

Problem definition

States: $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t \in \mathcal{R}^n$

Actions: $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathcal{R}^m$

Measurements: $\mathbf{z}_1, \ldots, \mathbf{z}_t \in \mathcal{R}^k$

Algorithm: $\mathbf{u}_{t+1} = \pi(\mathbf{z}_{1:t}, \mathbf{u}_{1:t})$

Rewards: $r_t = r(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{x}_t) \in \mathcal{R}$

Criterion: $J_\pi = \mathbb{E}_{\tau \sim \pi}\{ \sum_{r_t \sim \tau} \gamma^t r_t \} \in \mathcal{R}$

Goal: $\pi^* = \arg\max_\pi J_\pi$

Examples:

- Balancing pendulum (z=x: joint angles/velocities/accelerations, u: base vel., r: height)
- RC helicopter tricks (z: body pose+angles/vel/acc, u: motor torques, r: ???)
- Agent playing Atari games (z: RGB, u: joystick, r: score/lives, x: ???)

Problem definition

States: $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t \in \mathcal{R}^n$

Actions: $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathcal{R}^m$

Measurements: $\mathbf{z}_1, \ldots, \mathbf{z}_t \in \mathcal{R}^k$

Algorithm: $\mathbf{u}_{t+1} = \pi(\mathbf{z}_{1:t}, \mathbf{u}_{1:t})$

Rewards: $r_t = r(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{x}_t) \in \mathcal{R}$

Criterion: $J_\pi = \mathbb{E}_{\tau \sim \pi}\{ \sum_{r_t \sim \tau} \gamma^t r_t \} \in \mathcal{R}$

Goal: $\pi^* = \arg\max_\pi J_\pi$

Examples:

x: ???

z:

Problem definition

States: $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t \in \mathcal{R}^n$

Actions: $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathcal{R}^m$

Measurements: $\mathbf{z}_1, \ldots, \mathbf{z}_t \in \mathcal{R}^k$

Algorithm: $\mathbf{u}_{t+1} = \pi(\mathbf{z}_{1:t}, \mathbf{u}_{1:t})$

Rewards: $r_t = r(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{x}_t) \in \mathcal{R}$

Criterion: $J_\pi = \mathbb{E}_{\tau \sim \pi}\{ \sum_{r_t \sim \tau} \gamma^t r_t \} \in \mathcal{R}$

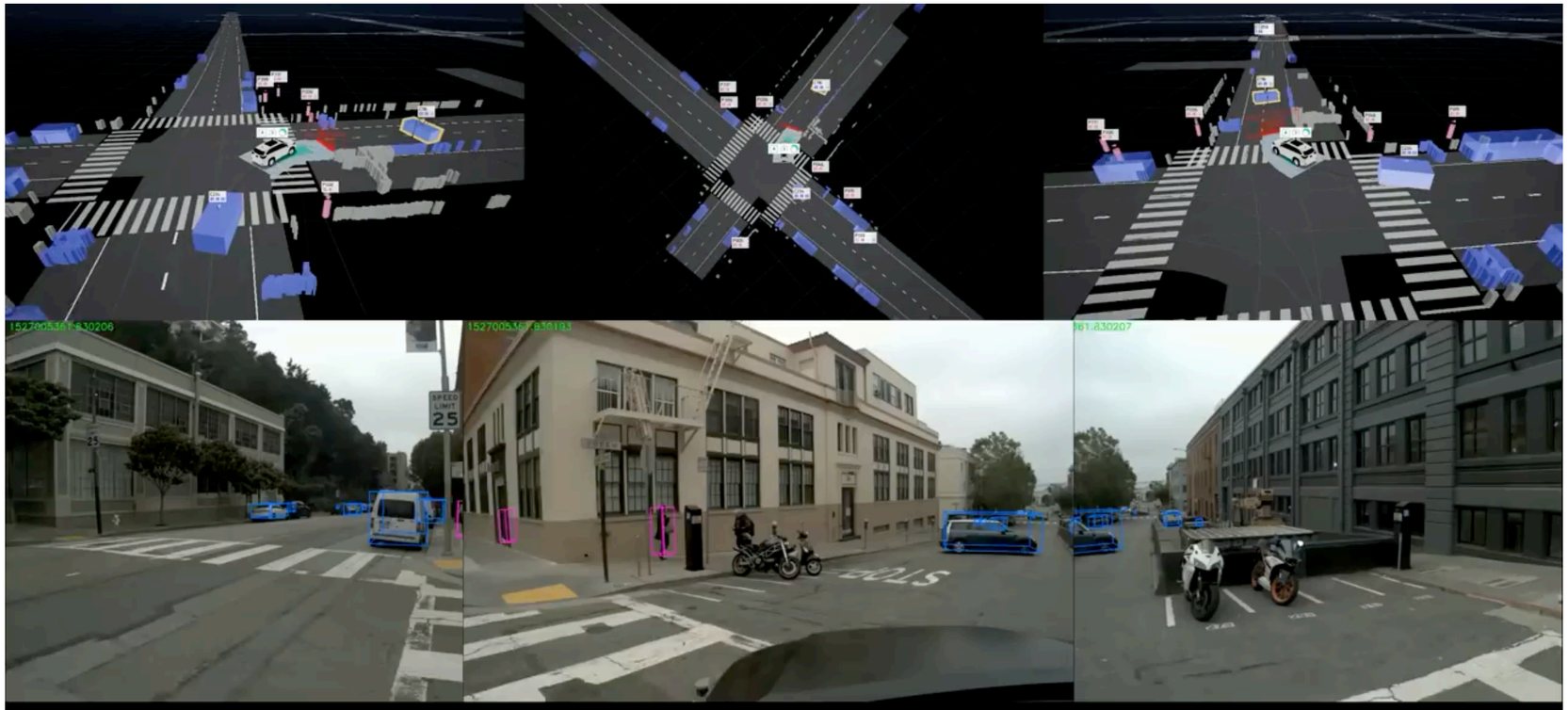
Goal: $\pi^* = \arg\max_\pi J_\pi$

- Autonomous car (z: RGB+lidar+IMU, u: wheel+throttle, r: reach goal+survive, x: ???)

Examples:

x:

z:



Problem definition

States: $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t \in \mathcal{R}^n$

Actions: $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathcal{R}^m$

Measurements: $\mathbf{z}_1, \ldots, \mathbf{z}_t \in \mathcal{R}^k$

Algorithm: $\mathbf{u}_{t+1} = \pi(\mathbf{z}_{1:t}, \mathbf{u}_{1:t})$

Rewards: $r_t = r(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{x}_t) \in \mathcal{R}$

Criterion: $J_\pi = \mathbb{E}_{\tau \sim \pi}\{ \sum_{r_t \sim \tau} \gamma^t r_t \} \in \mathcal{R}$

Goal: $\pi^* = \arg\max_\pi J_\pi$

Examples:

- Balancing pendulum (z=x: joint angles/velocities/accelerations, u: base vel., r: height)
- RC helicopter tricks (z: body pose+angles/vel/acc, u: motor torques, r: ???)
- Agent playing Atari games (z: RGB, u: joystick, r: score/lives, x: ???)
- Autonomous car (z: RGB+lidar+IMU, u: wheel+throttle, r: reach goal+survive, x: ???)

Problem definition

States: $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t \in \mathcal{R}^n$

Actions: $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathcal{R}^m$

Measurements: $\mathbf{z}_1, \ldots, \mathbf{z}_t \in \mathcal{R}^k$

Algorithm: $\mathbf{u}_{t+1} = \pi(\mathbf{z}_{1:t}, \mathbf{u}_{1:t})$

Rewards: $r_t = r(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{x}_t) \in \mathcal{R}$

Criterion: $J_\pi = \mathbb{E}_{\tau \sim \pi}\{ \sum_{r_t \sim \tau} \gamma^t r_t \} \in \mathcal{R}$

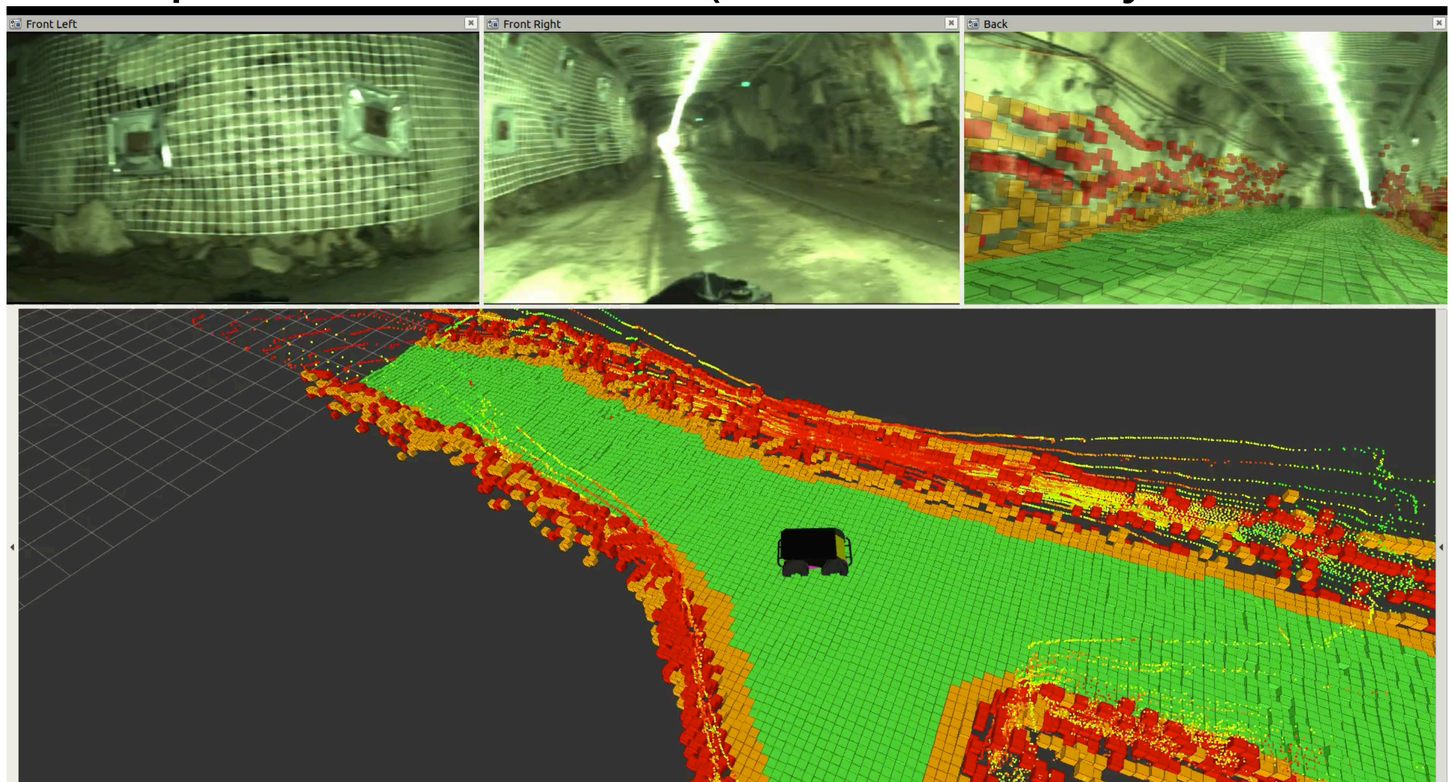
Goal: $\pi^* = \arg\max_\pi J_\pi$

- DARPA exploration scenario (x: successively constructed map+pose)

Examples:

z:

x:



Problem definition

States: $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t \in \mathcal{R}^n$

Actions: $\mathbf{u}_1, \ldots, \mathbf{u}_t \in \mathcal{R}^m$

Measurements: $\mathbf{z}_1, \ldots, \mathbf{z}_t \in \mathcal{R}^k$

Algorithm: $\mathbf{u}_{t+1} = \pi(\mathbf{z}_{1:t}, \mathbf{u}_{1:t})$

Rewards: $r_t = r(\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{x}_t) \in \mathcal{R}$

Criterion: $J_\pi = \mathbb{E}_{\tau \sim \pi}\{ \sum_{r_t \sim \tau} \gamma^t r_t \} \in \mathcal{R}$

Goal: $\pi^* = \arg\max_\pi J_\pi$

Examples:

- Balancing pendulum (z=x: joint angles/velocities/accelerations, u: base vel., r: height)
- RC helicopter tricks (z: body pose+angles/vel/acc, u: motor torques, r: ???)
- Agent playing Atari games (z: RGB, u: joystick, r: score/lives, x: ???)
- Autonomous car (z: RGB+lidar+IMU, u: wheel+throttle, r: reach goal+survive, x: ???)
- DARPA exploration scenario (x: successively constructed map+pose)
- Exotic tasks also covered: e.g. Active SLAM (r: accurate state estimate)

Summary

- State, action, measurements, reward, cost, criterion,
- Goal is policy/algorithmn/pipeline/regulator that optimise criterion
- Two subproblems: Perception vs motion planning/control
- Further reading Probabilistic Robotics book https://docs.ufpr.br/~danielsantos/ProbabilisticRobotics.pdf
- Next lecture: Localization