

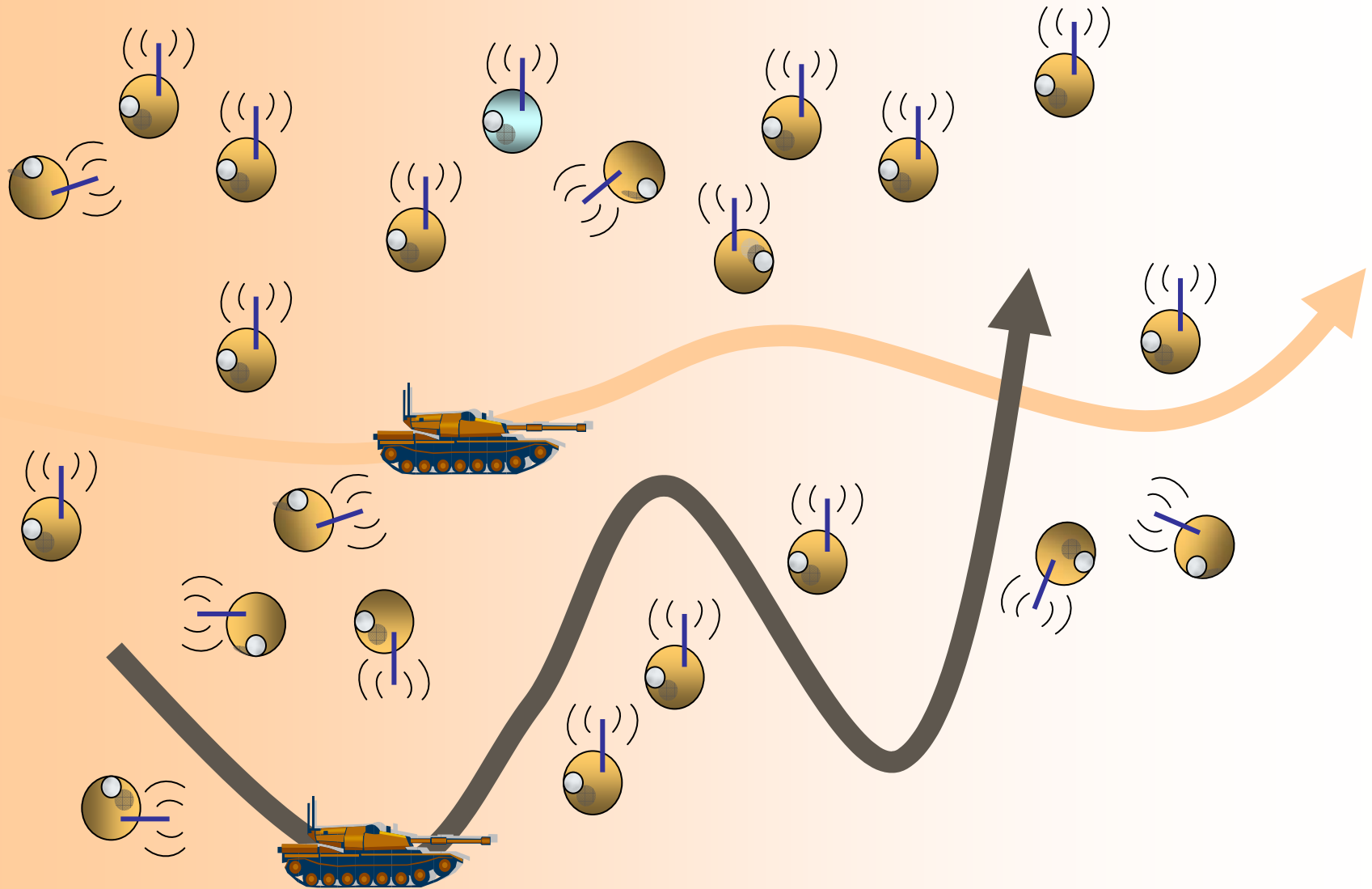
# **Adaptive Sensor Management: POMDP Approximation Methods**

**Edwin K. P. Chong**

**ECE and Math  
Colorado State University**

**ARO-MURI Adaptive Sensing and Waveform Design Workshop  
Aug. 2-3, 2005**

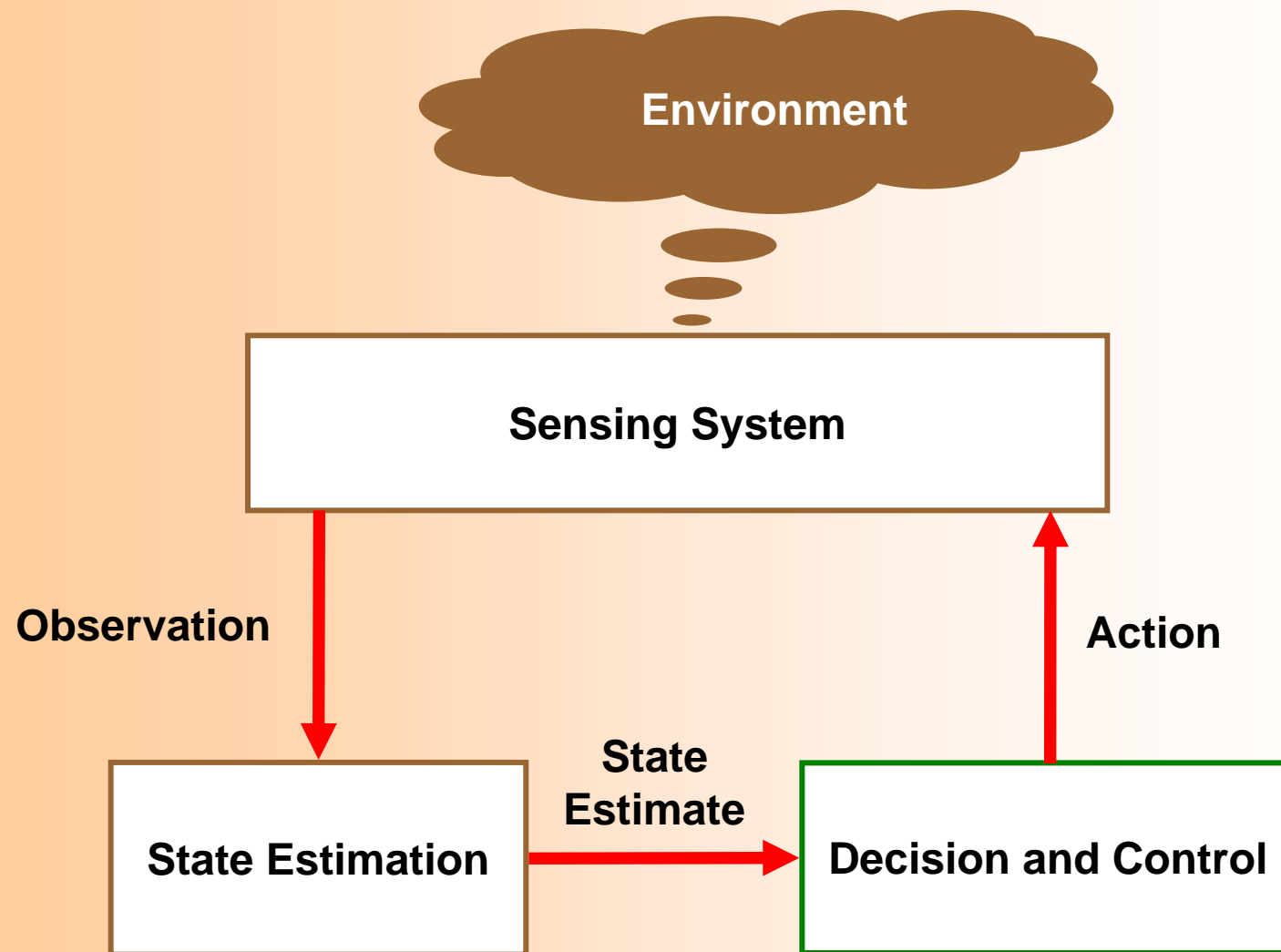
# Adaptive Sensor Management



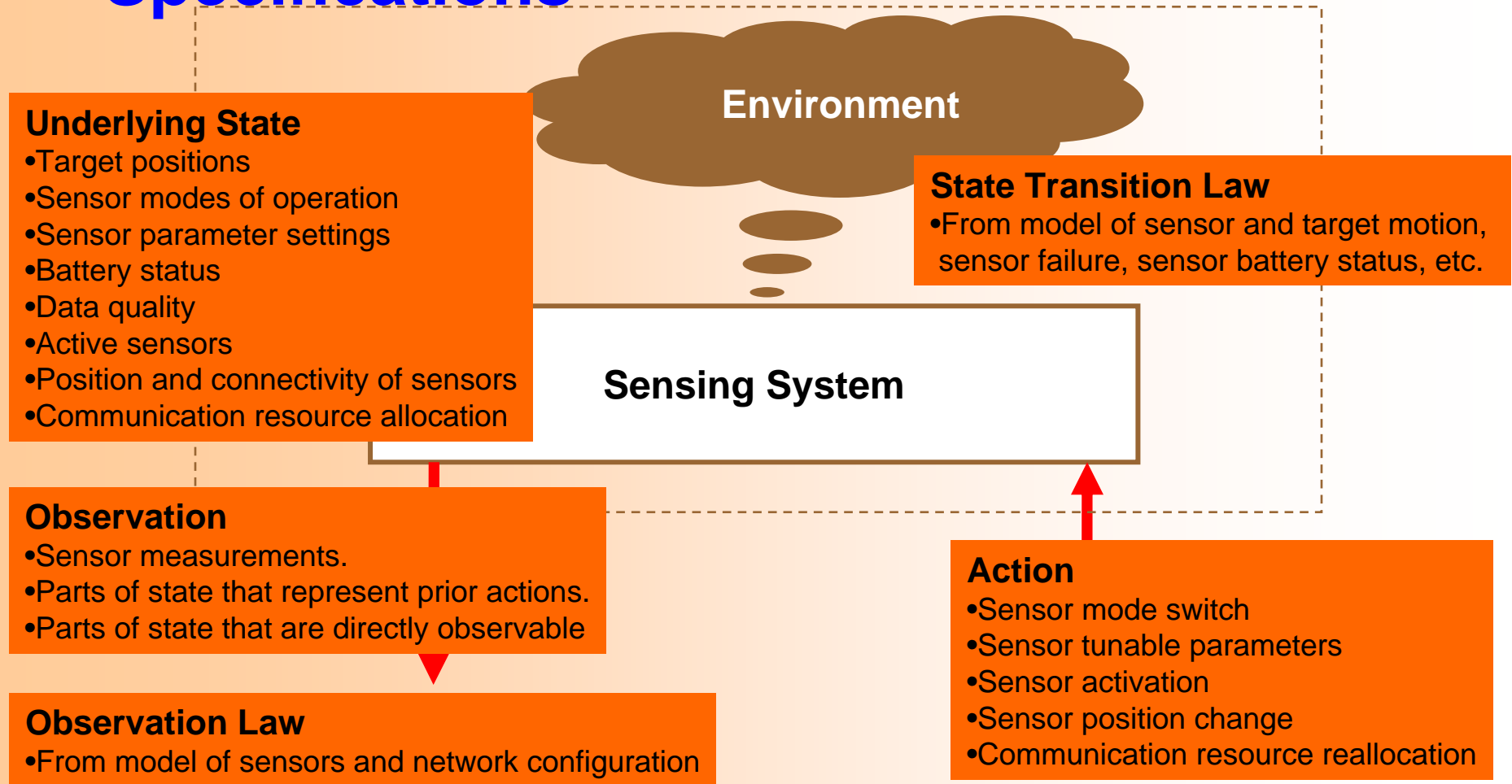
## Basic Approach

- Formulate sensor management problem as a partially observable Markov decision process (POMDP).
- Apply Q-value approximation methods.
- Combine in receding horizon lookahead.
- “Non-myopic.”
- Special architecture for Monte Carlo sampling methods.

# Basic Framework



# Sensor Management Plant Specifications

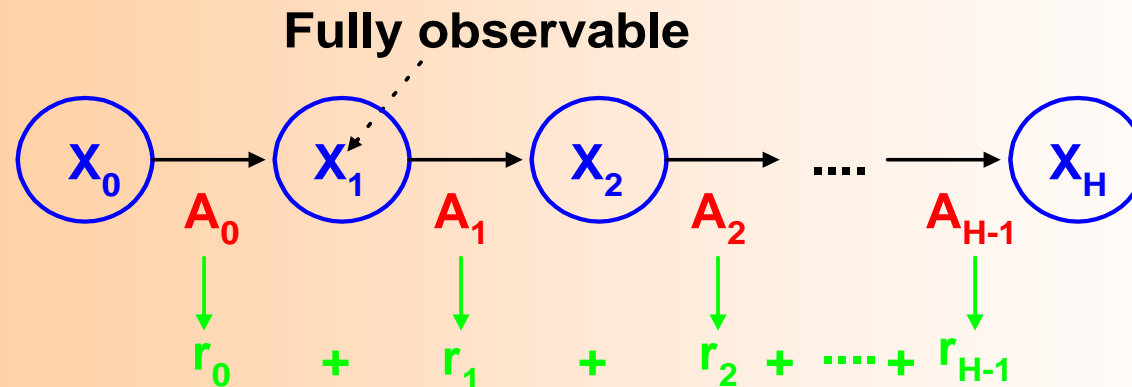


# Markov Decision Process (MDP)

- Ingredients:
  - ▶ System state  $x$
  - ▶ Control action
  - ▶ Reward  $R(x, a)$
  - ▶ State-transition probability  $T(y | x, a)$
- Optimization criterion:
  - ▶  $V_H(x_0) = E[R(x_0, a_0) + \dots + R(x_{H-1}, a_{H-1}) | x_0]$
- Find sequence of actions to maximize expected total reward (over a large horizon  $H$ ).
- Action can depend on state  $\Rightarrow$  **policy**.

# Policy

- **Policy:** mapping from states to actions (state feedback).
- Following a policy means:



- **Goal:** a policy that maximizes the expected total reward.

# Optimal Policy

- Objective:
  - ▲  $V_H^*(x_0) = \max E[R(x_0, a_0) + \dots + R(x_{H-1}, a_{H-1}) \mid x_0]$   
where the max is over all policies,  $a_i = \pi_i(x_i)$ ,  
and  $\pi_i$  is policy at time  $i$ .
- For large  $H$ ,  $\pi_i$  independent of  $i$ .
  - ▲ Stationary policy

# Optimal Policy (Cont'd)

- Define  $Q(x, a) = R(x, a) + E[V_{H-1}^*(y)|x, a]$ 
  - ▲ “Utility” of action  $a$  at state  $x$ .
  - ▲ Name: **Q-value** of action  $a$  at state  $x$ .
- Key identities (Bellman’s equations):
  - ▲  $V_H^*(x) = \max_a Q(x, a)$
  - ▲  $\pi_0^*(x) = \operatorname{argmax}_a Q(x, a)$

# Partially Observable MDP (POMDP)

- Generalization of MDP
  - ▶ Remove the full observability of states.
  - ▶ Add set of **observations** to MDP.
  - ▶ Each state-action pair gives an observation distribution  $O(z | x, a)$ .
  - ▶ Observation provides **hint** about underlying state.
- Can keep track of **belief state** (posterior prob. distribution over states) over time.  
Also called **information state**.
- **Policy**: mapping from belief states to actions.
- Turns out that if we treat belief states as states, we have an MDP. So for now we will not distinguish POMDPs from MDPs.

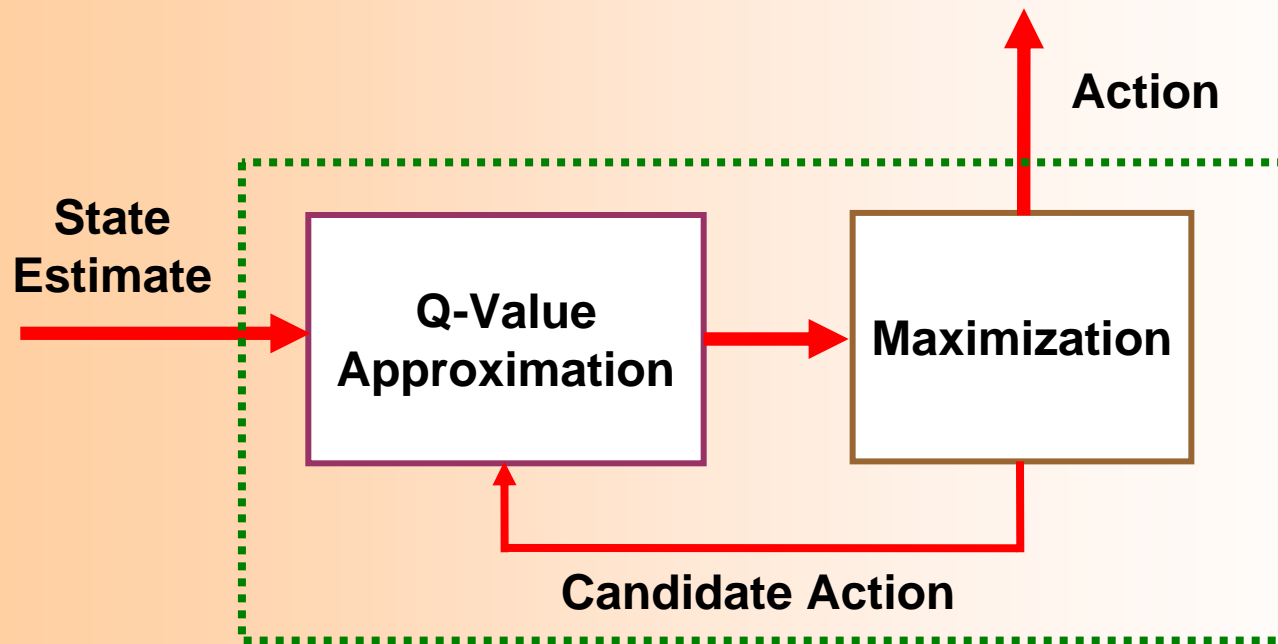
# Approximation Methods

- Recall:
  - ▶  $V_H^*(x) = \max_a Q(x, a)$   
 $= \max_a R(x, a) + E[V_{H-1}^*(y)|x, a]$
  - ▶  $\pi_0^*(x) = \operatorname{argmax}_a Q(x, a)$
- Q-value depends on optimal policy!
- Two-pronged solution approach:
  - ▶ Approximate Q-value
  - ▶ Apply receding-horizon method

# Receding-Horizon Control

- For large horizon  $H$ , policy is approximately stationary.
- At each time, if state is  $x$ , then apply action
$$\begin{aligned}\pi^*(x) &= \operatorname{argmax}_a Q(x,a) \\ &= \operatorname{argmax}_a R(x,a) + E[V_{H-1}^*(y)|x,a]\end{aligned}$$
- Compute estimate of Q-value at each time.
- Policy computation: perform optimization at each decision epoch (same burden as myopic approach!).

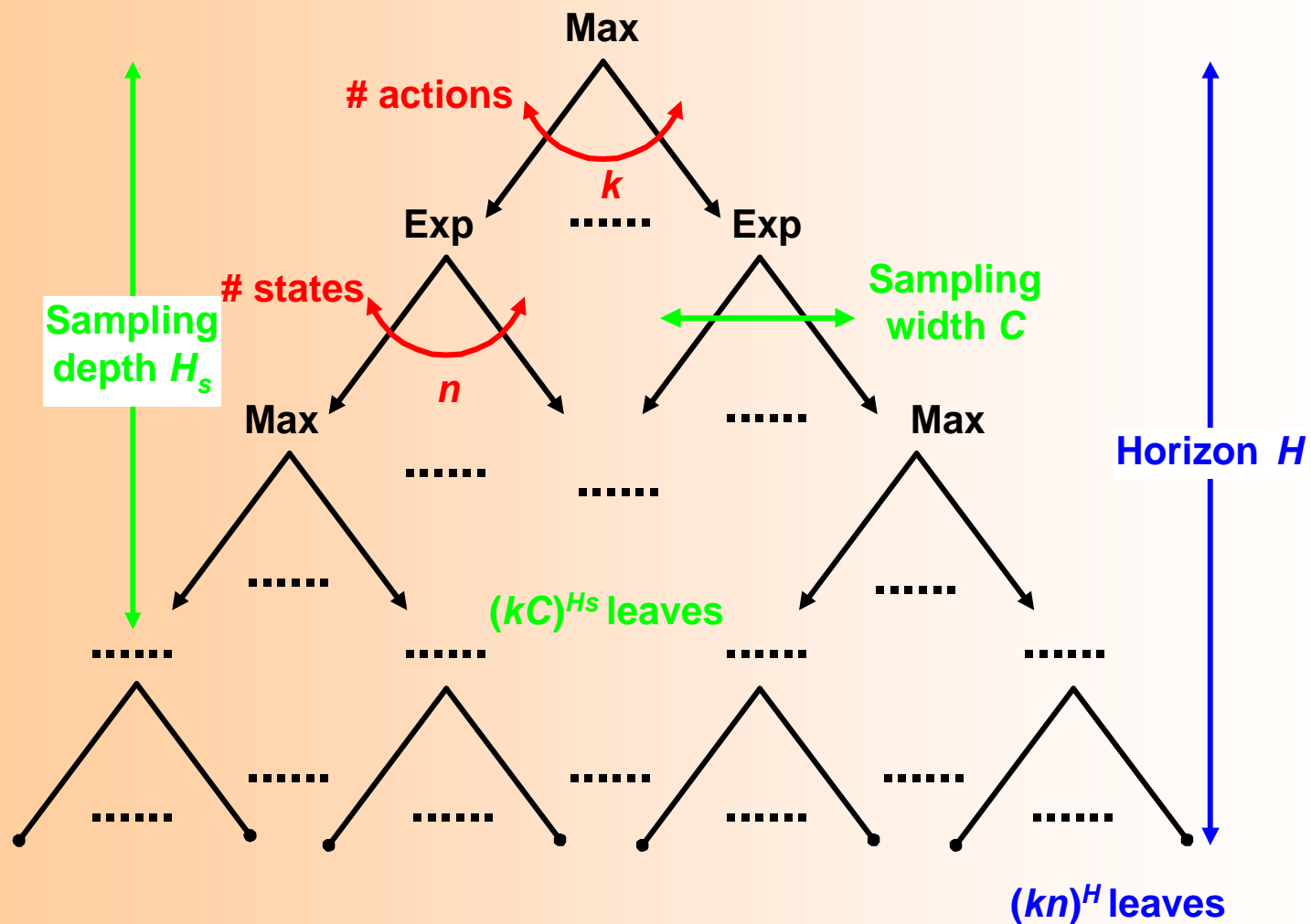
# Decision and Control Module



# Approximating Q-Values

- Relaxation
- Reduction to classification
- Heuristic approximation
- Parametric function approximation
  - ▲ Neurodynamic programming
  - ▲ Q-learning
- Monte Carlo sampling:
  - ▲ KMN-sampling
  - ▲ Policy rollout
  - ▲ Hindsight optimization

# KMN-sampling



## KMN-sampling (Cont'd)

- For a given desired accuracy, how large should sampling width and depth be?
- Answer due to **Kearns, Mansour, and Ng** (1999).
- For reasonable accuracy, required sampling width and depth are prohibitively large.
- In practice, cannot guarantee desired accuracy.

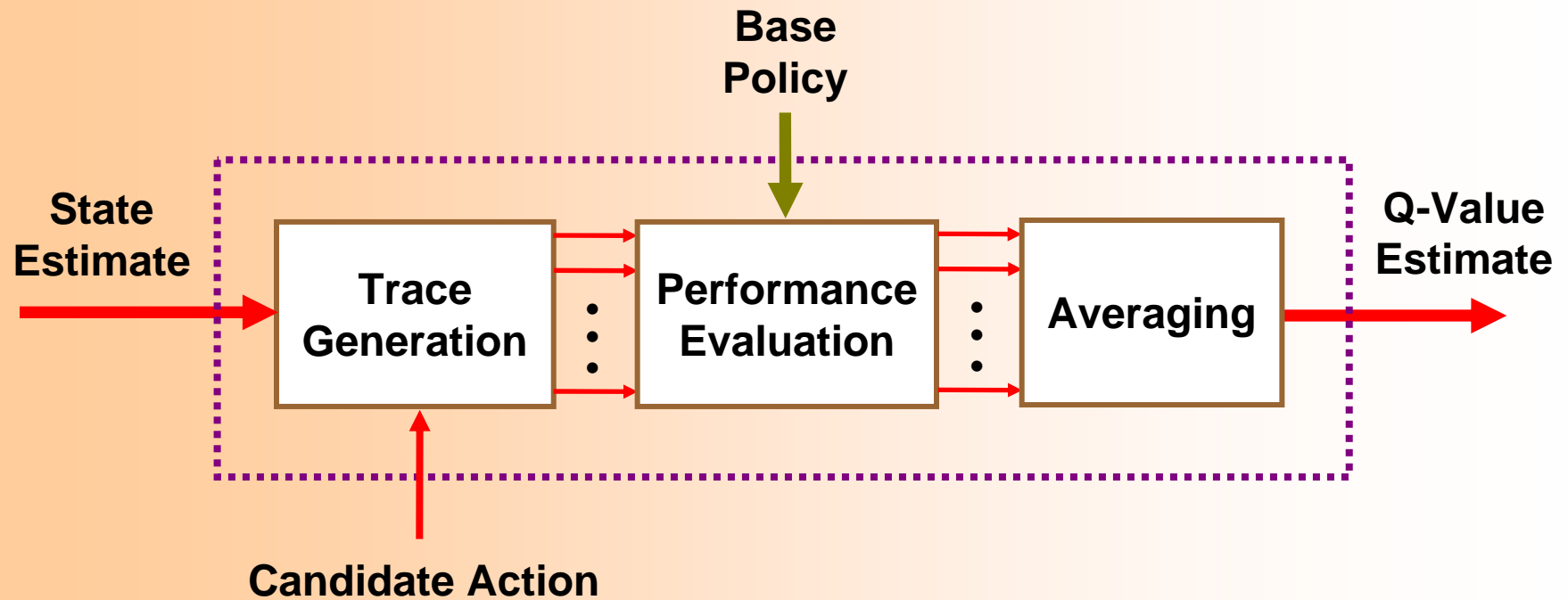
# Policy Rollout

- Due to **Castanon, Bertsekas, et al.**
- Recall:  $Q(x,a) = R(x,a) + E[V_{H-1}^*(y)|x,a]$   
 $= R(x,a) + E[\max_{\pi} V_{H-1}^{\pi}(y)|x,a]$

where  $V_{H-1}^{\pi}(y)$  is the value of following policy  $\pi$ .

- Given a policy  $\pi$  (**base policy**), use  
 $R(x,a) + E[V_{H-1}^{\pi}(y)|x,a]$   
as an estimate of Q-value.
- Gives a **lower bound** to true Q-value.

# Policy Rollout (Cont'd)



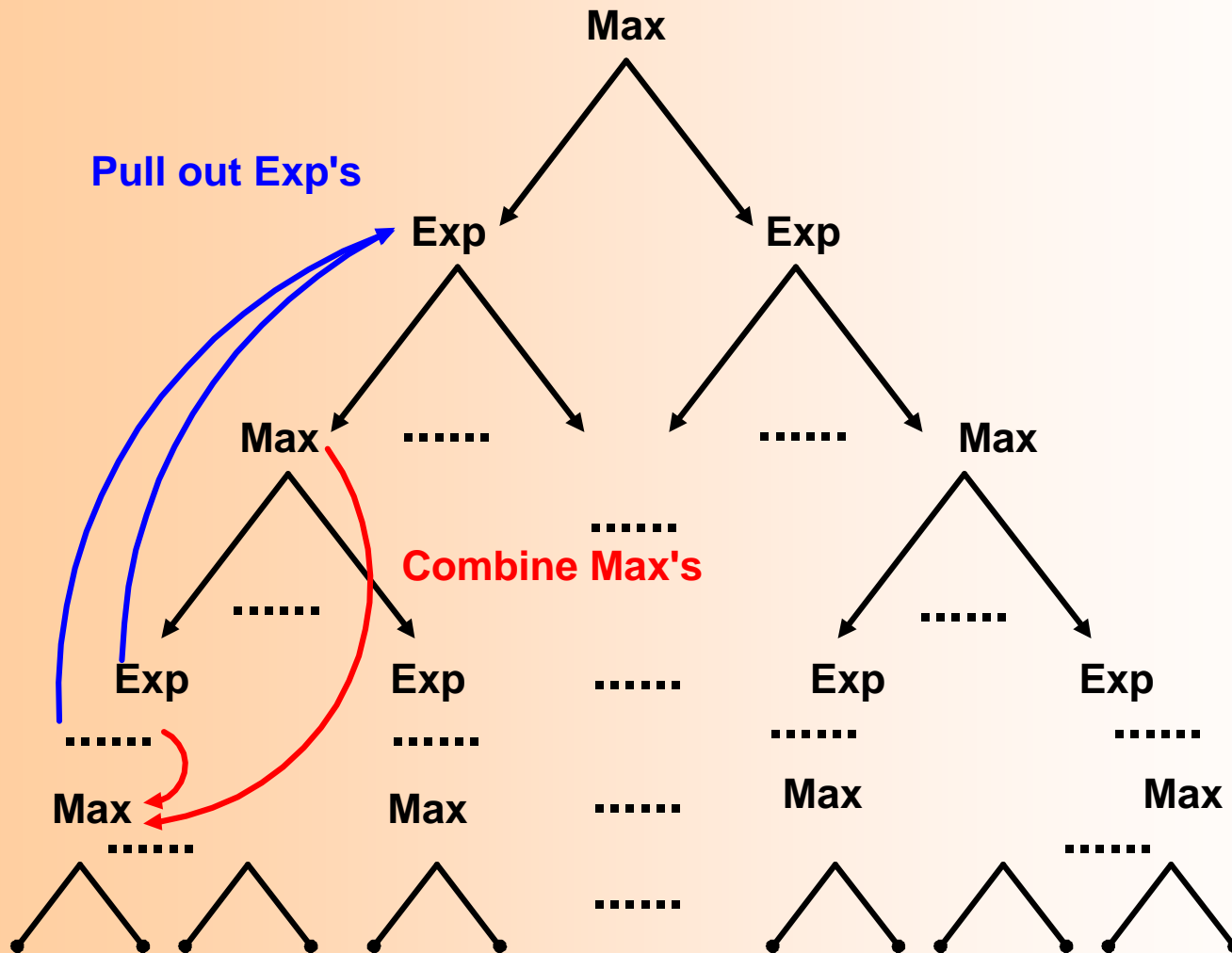
# Parallel Policy Rollout

- Generalization of policy rollout, due to **Chang, Givan, and Chong**.
- Given a set  $\Pi$  of base policies, use
$$R(x,a) + E[\max_{\pi \in \Pi} V_{H-1}^{\pi}(y)|x,a]$$
as an estimate of Q-value.
- This estimate is more accurate than policy rollout.
- Still gives a **lower bound** to true Q-value.

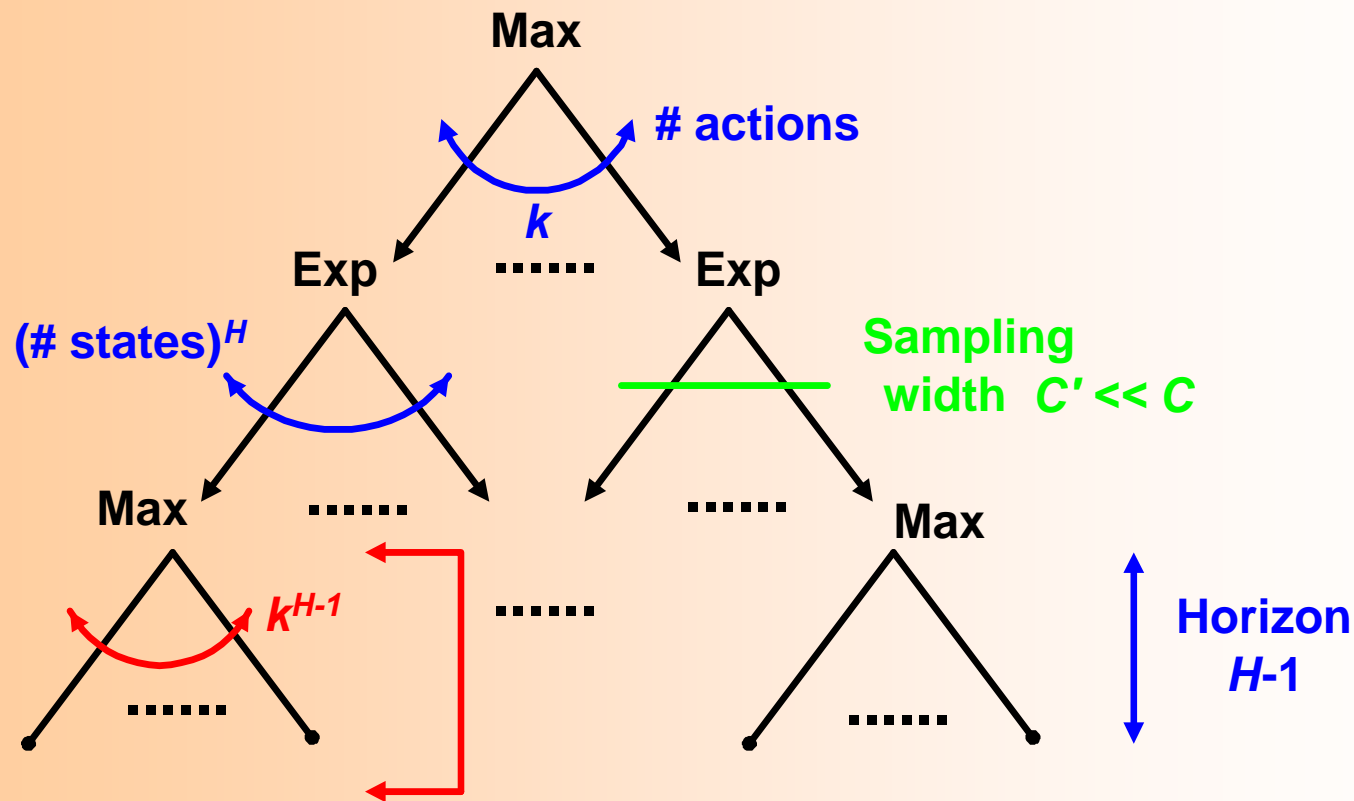
# Hindsight Optimization

- Due to **Chang, Givan, and Chong**.
- Recall:  $Q(x, a) = R(x, a) + E[\max_{\pi} V_{H-1}^{\pi}(x_1) | x, a]$   
where  $V_{H-1}^{\pi}(x_1) = E[R(x_1, \pi(x_1)) + \dots$   
 $+ R(x_{H-1}, \pi(x_{H-1})) | x_1]$
- Use  
 $R(x, a) + E[\max R(x_1, a_1) + \dots + R(x_{H-1}, a_{H-1}) | x]$   
as an estimate of Q-value, where the  
max is over all action sequences  $a_1, \dots, a_{H-1}$ .
- Equivalent to exchanging  
max and expectation.
- Gives an **upper bound** to true Q-value.

# Hindsight Optimization (Cont'd)

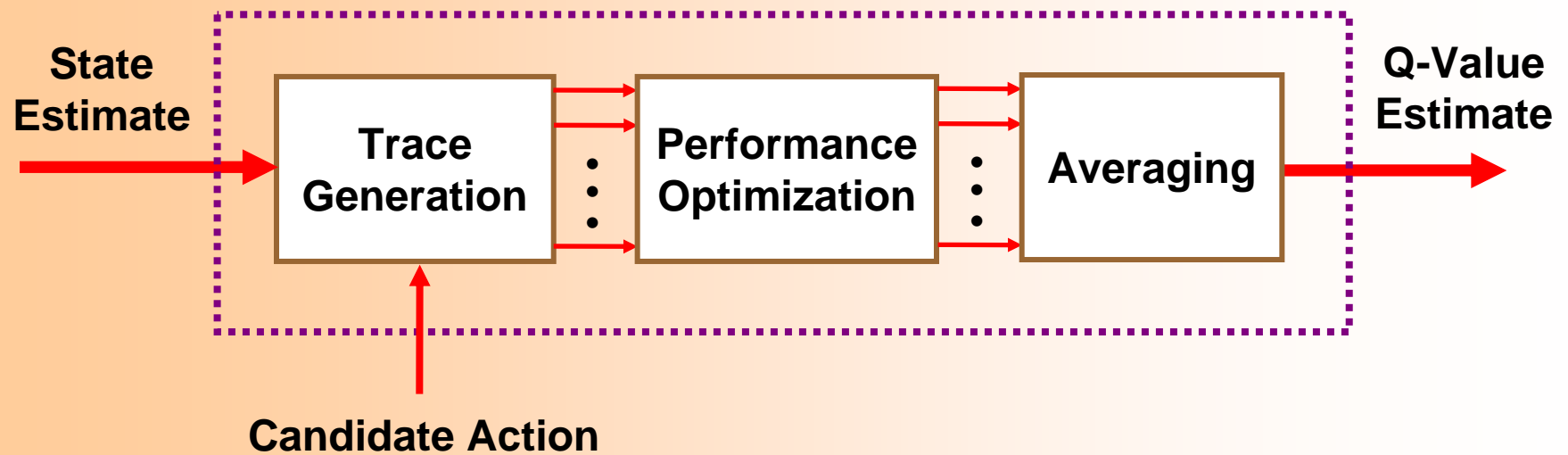


# Hindsight Optimization (Cont'd)

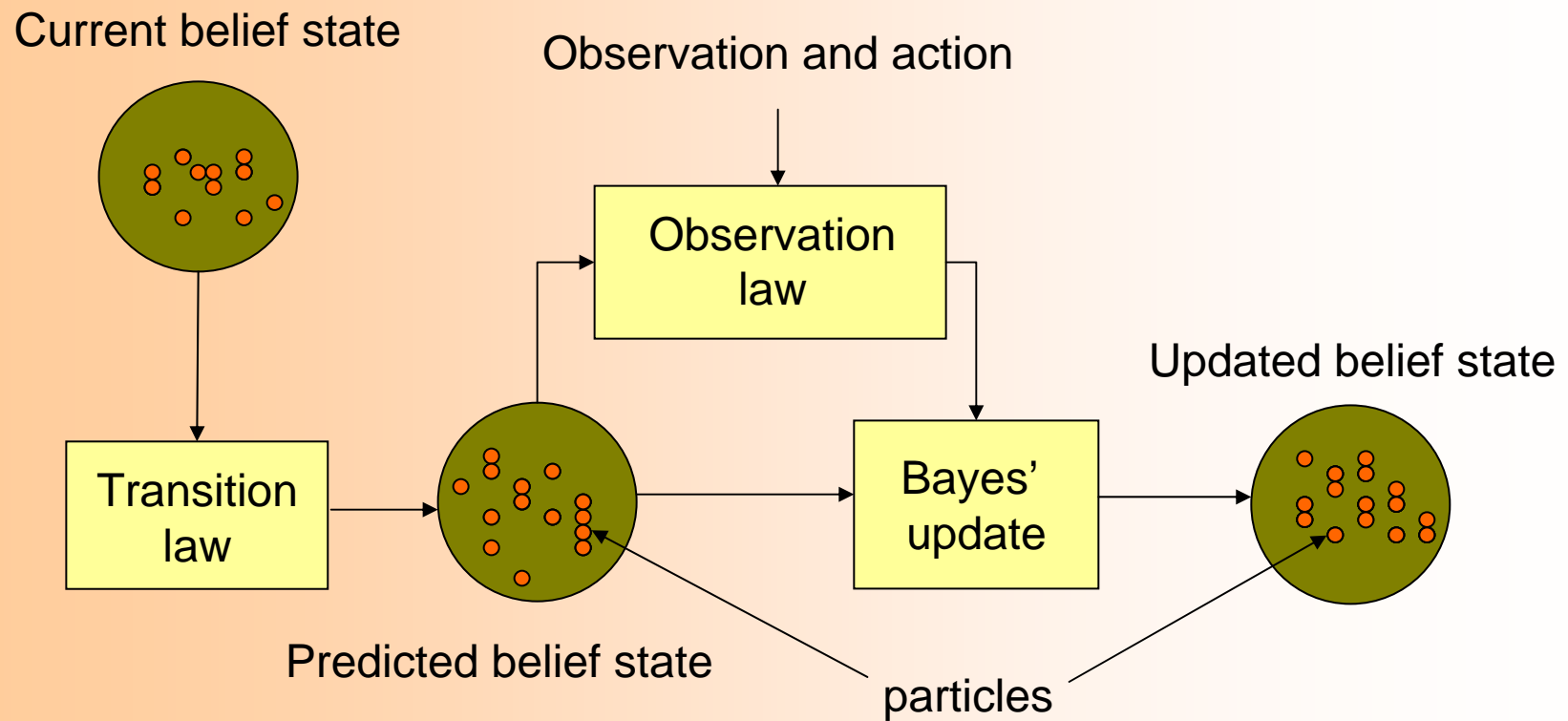


Selecting best action seq. from  $k^{H-1}$  choices:  
an off-line optimization problem.

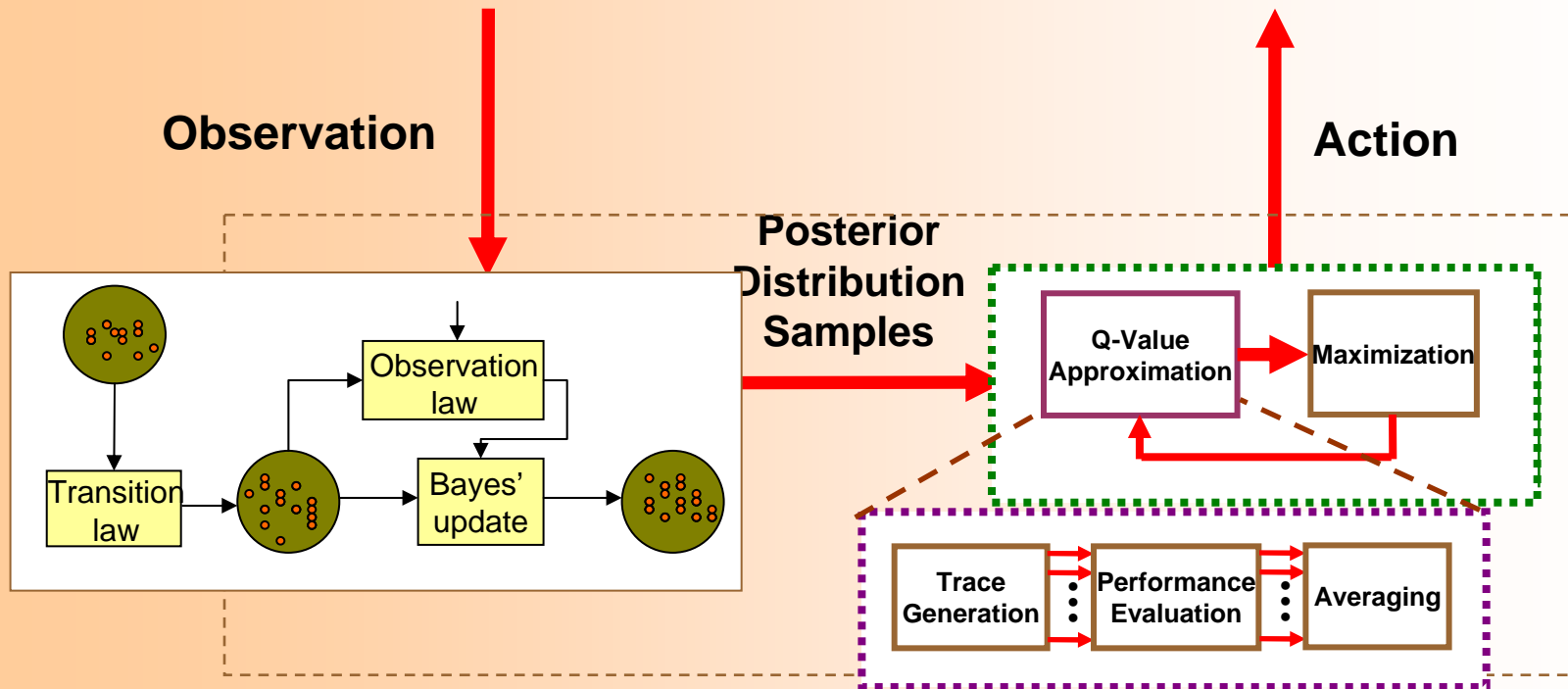
# Hindsight Optimization (Cont'd)



# Belief State Update: Particle Filter



# Monte Carlo Sensor Manager



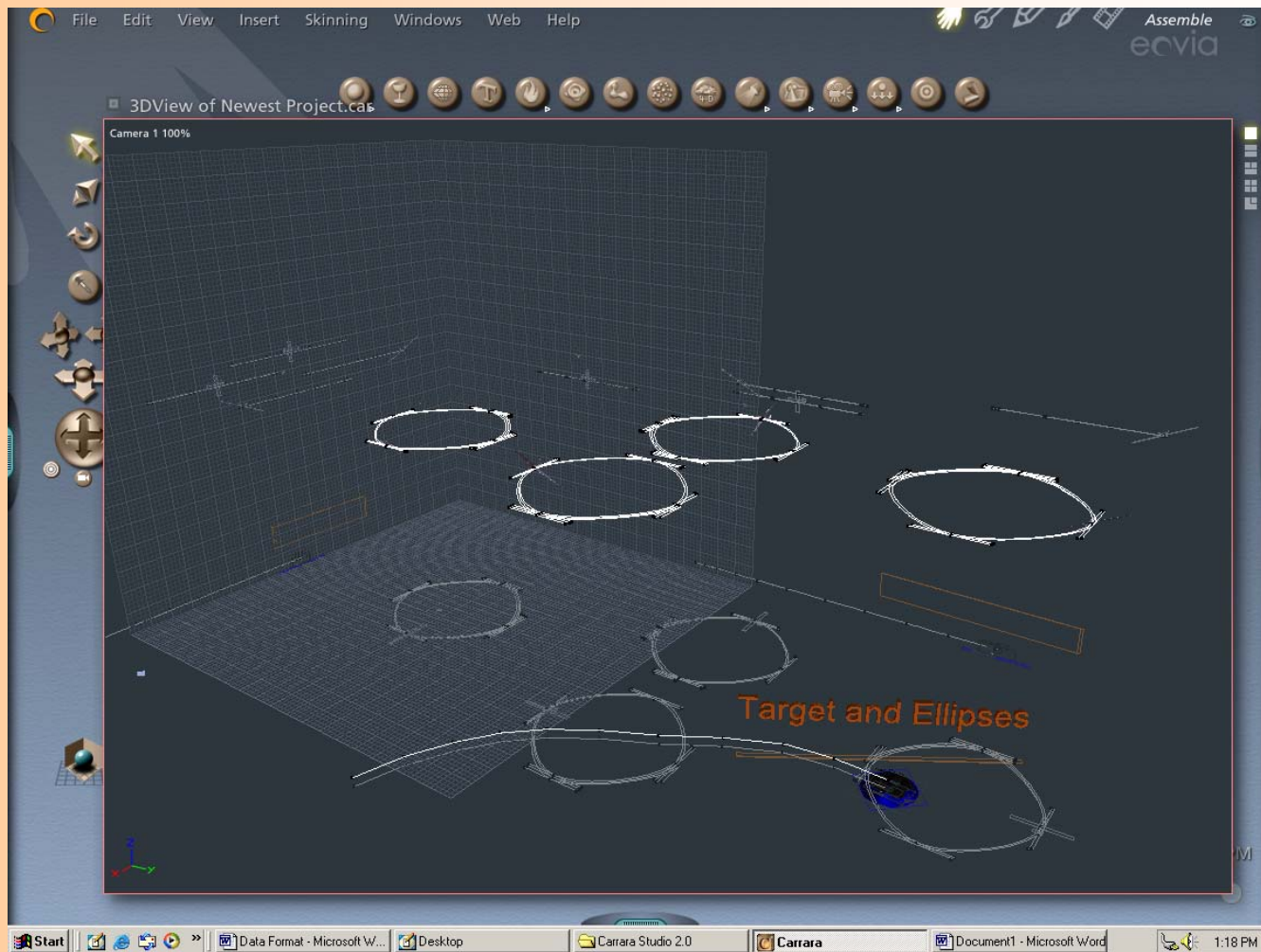
# Advantages of Monte Carlo Approach

- **Flexible:** A variety of sensor management scenarios can be tackled using the same framework.
- **General:** Does not require analytical tractability.
- **Modular:** Models of individual system components (e.g., sensor types, target motion) may be treated as "plug-in" modules.
- **Portable:** Integrates with existing simulators.
- **Non-Myopic:** Allows trading off short-term for long-term gains.

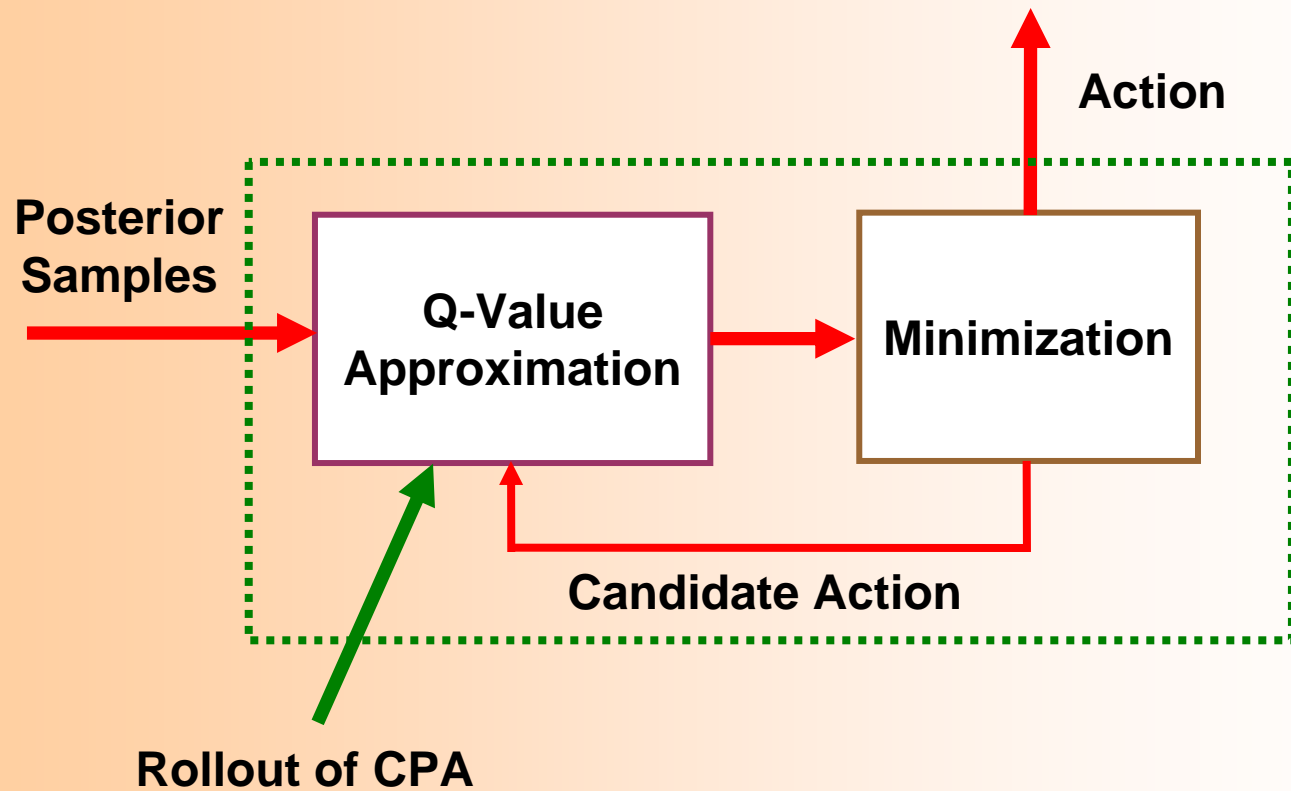
# Example Scenario

- UAVs in circular trajectories.
- Radar in each UAV, nonhomogeneous costs.
- Sensor scheduling problem:  
dynamically select which radar to activate.
- Single ground target.
- Compare CPA with POMDP scheduler.
- Objective function: Cumulative tracking error plus sensor usage cost.

# Physical Configuration



# Decision and Control Module



## Modified (CO) Rollout

- CO base policy: CPA based on actual target position.
  - ▲ Select sensor that is closest to target.
- At each decision epoch, starting from each particle, compute cumulative cost of CPA over horizon.
- Take average of above costs as Q-value approximation.
- No need to keep track of belief state in future.

# Rendered Animation

- Animation of actual run of CPA and POMDP+PF simulators.



# Numerical Comparison

- Cumulative cost:
  - ▲ CPA: 230
  - ▲ POMDP: 47
- Cumulative tracking error:
  - ▲ CPA: 87
  - ▲ POMDP: 10

# Rendered Animation



# Questions?



- Contact information:
  - ▶ [echong@engr.colostate.edu](mailto:echong@engr.colostate.edu)
  - ▶ Dept. of Electrical and Computer Engineering  
**Colorado State University**  
Fort Collins, CO 80523-1373
  - ▶ Tel: 970-491-7858
  - ▶ Fax: 970-491-2249