ADP advances for real-world Building Control

Paul Beuchat
Angelos Georghiou, John Lygeros

Automatic Control Laboratory, ETH Zürich

September 6 2016

Motivation

Attributes of Approximate Dynamic Programming (ADP):

- If a good approximation can be achieved, then the method can be very powerful
- As shown in Deliverable 4.1.2, ADP is amenable to distributed control
- Many learning algorithms and their theory are based on ADP
- PCOA is also based on the theory of ADP

Motivation

Attributes of Approximate Dynamic Programming (ADP):

- If a good approximation can be achieved, then the method can be very powerful
- As shown in Deliverable 4.1.2, ADP is amenable to distributed control
- Many learning algorithms and their theory are based on ADP
- PCOA is also based on the theory of ADP

Motivation

Attributes of Approximate Dynamic Programming (ADP):

- If a good approximation can be achieved, then the method can be very powerful
- As shown in Deliverable 4.1.2, ADP is amenable to distributed control
- Many learning algorithms and their theory are based on ADP
- PCOA is also based on the theory of ADP

Motivation

Attributes of Approximate Dynamic Programming (ADP):

- If a good approximation can be achieved, then the method can be very powerful
- As shown in Deliverable 4.1.2, ADP is amenable to distributed control
- Many learning algorithms and their theory are based on ADP
- PCOA is also based on the theory of ADP

Outline

Outline

Approxime LP and Policy

A brief recap of our algorithm

Approxime LP and Policy

A brief recap of our algorithm

Choose a function space: $Q \in \hat{\mathcal{F}}(\mathcal{X} \times \mathcal{U})$

Approxime LP and Policy

A brief recap of our algorithm

Choose a function space: $Q \in \hat{\mathcal{F}}(\mathcal{X} \times \mathcal{U})$

Solve the following optimisation:

$$\hat{Q}^*(x,u) = \max_{Q} \int_{\mathcal{X}\times\mathcal{U}} Q(x,u)\ c(d(x,u))$$

$$\text{s.t.} \quad Q \in \hat{\mathcal{F}}(\mathcal{X} \times \mathcal{U})$$

$$Q(x,u) \leq FQ(x,u)\,, \forall\, x \in \mathcal{X}\,,\ u \in \mathcal{U}$$

Approxime LP and Policy

A brief recap of our algorithm

Choose a function space: $Q \in \hat{\mathcal{F}}(\mathcal{X} \times \mathcal{U})$

Solve the following optimisation:

$$\begin{aligned} \hat{Q}^*(x,u) = \max_{Q} \quad & \int_{\mathcal{X}\times\mathcal{U}} Q(x,u)\; c(d(x,u)) \\ \text{s.t.} \quad & Q \in \hat{\mathcal{F}}(\mathcal{X} \times \mathcal{U}) \\ & Q(x,u) \leq FQ(x,u)\,, \forall\, x \in \mathcal{X}\,,\; u \in \mathcal{U} \end{aligned}$$

Use this policy to make online decisions

Approx. Policy: $\hat{\pi}(x_t) = \arg\min_{u\in\mathcal{U}} \hat{Q}^*(x,u)$

Approxime LP and Policy

A brief recap of our algorithm

Choose a function space: $Q \in \hat{\mathcal{F}}(\mathcal{X} \times \mathcal{U})$ *Quadratic*

Solve the following optimisation:

$$\hat{Q}^*(x,u) = \max_{Q} \int_{\mathcal{X}\times\mathcal{U}} Q(x,u)\ c(d(x,u))$$

$$\text{s.t.} \quad Q \in \hat{\mathcal{F}}(\mathcal{X} \times \mathcal{U})$$

$$Q(x,u) \leq FQ(x,u)\,, \quad \forall\, x \in \mathcal{X}\,,\ u \in \mathcal{U}$$

Use this policy to make online decisions

$$\text{Approx. Policy: } \hat{\pi}(x_t) = \arg\min_{u\in\mathcal{U}} \hat{Q}^*(x,u)$$

Performance Guarantees

Fitting Error (∞-norm based)

$$\left\| Q^* - \hat{Q}^* \right\|_{1,c(x,u)} \leq \frac{2}{1-\gamma^M} \min_{\hat{Q}\in\hat{\mathcal{F}}} \|Q^* - \hat{Q}\|_\infty$$

Fitting Error (Lyapunov function bases)

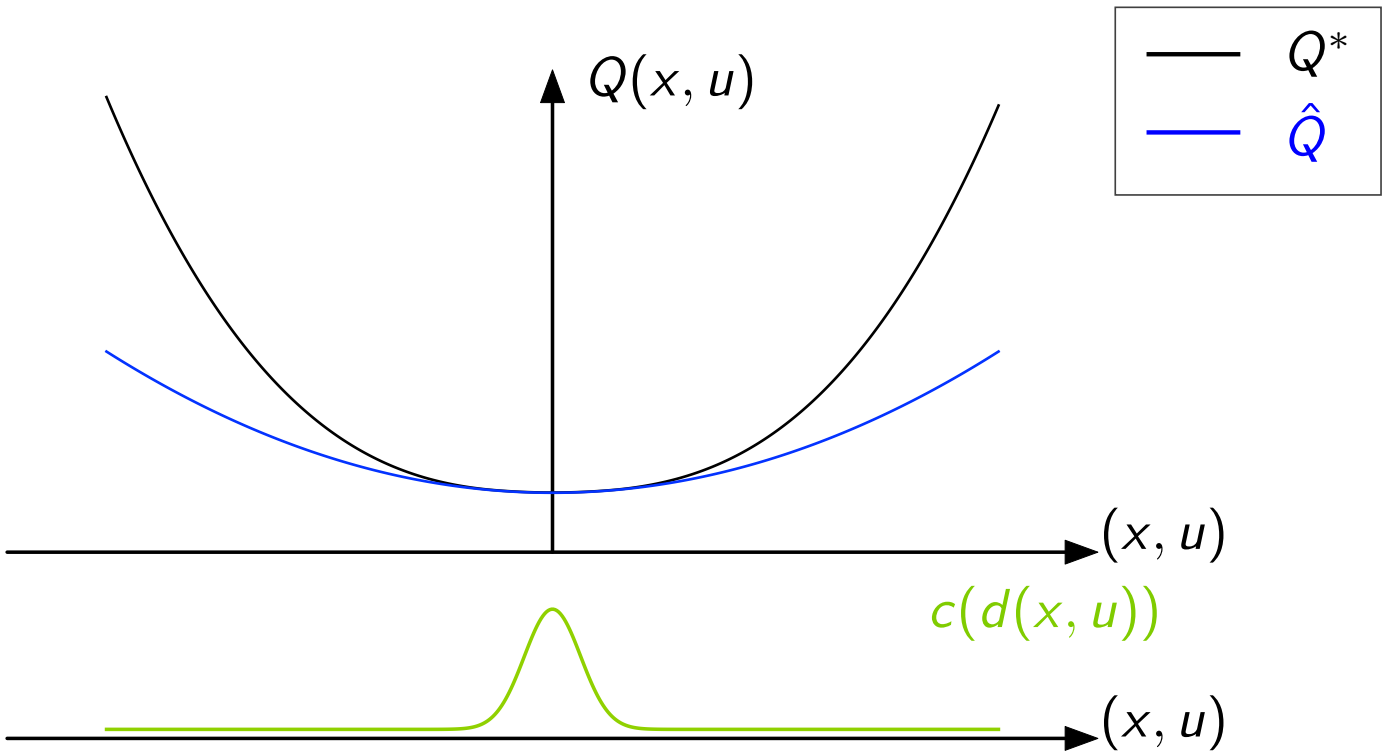
$$\left\|Q^* - \hat{Q}^*\right\|_{1,c(x,u)} \leq \frac{2\,\|\hat{Q}^+\|_{1,c(x,u)}}{1-\beta_{\hat{Q}^+}^M} \min_{\hat{Q}\in\hat{\mathcal{F}}} \left\|Q^* - \hat{Q}\right\|_{\infty,1/\hat{Q}^+}$$

Online Performance

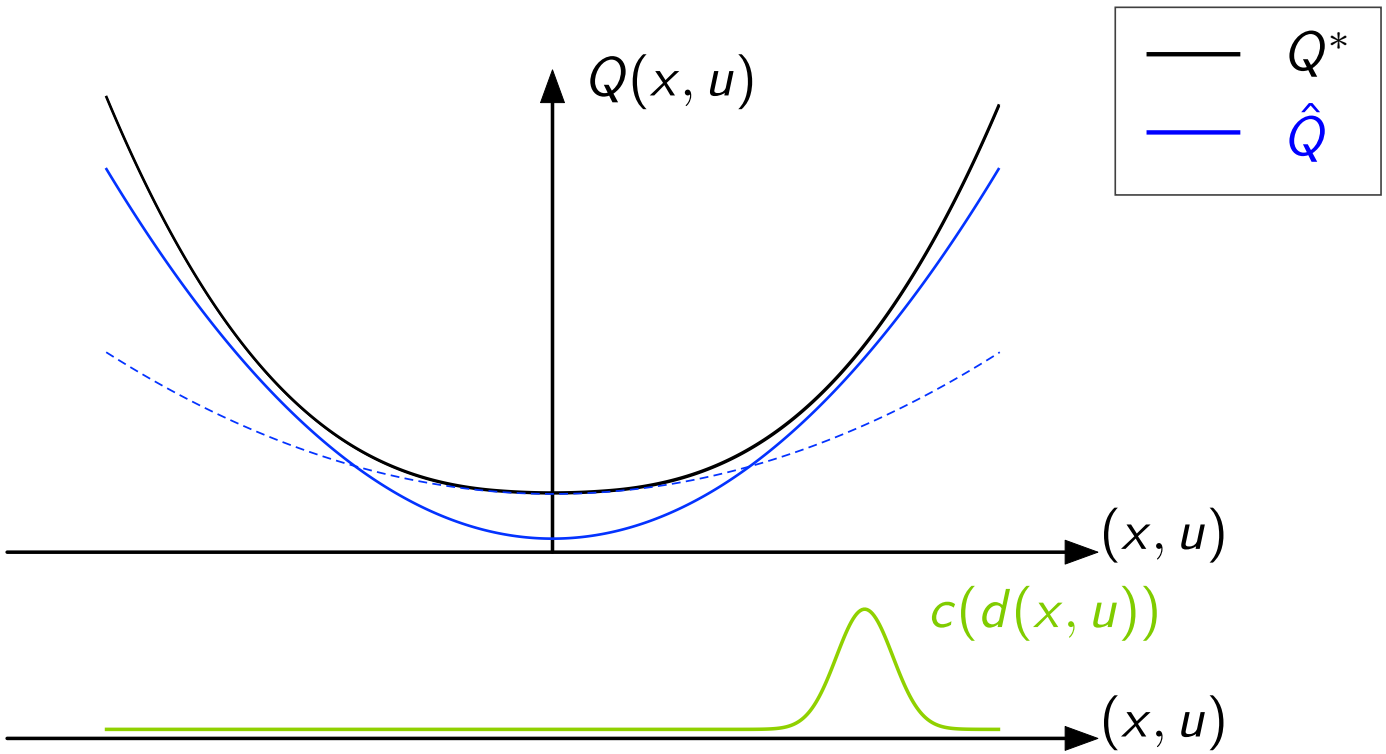
$$\| V_{\hat{\pi}} - V^* \|_{1,\nu} \leq \frac{1}{1-\gamma} \left\| Q^* - \left(F^{D-1}\hat{Q}\right) \right\|_{1,(1-\gamma)\tilde{\mu}_\nu^{\hat{\pi}}}$$

Outline

LP Approach to ADP

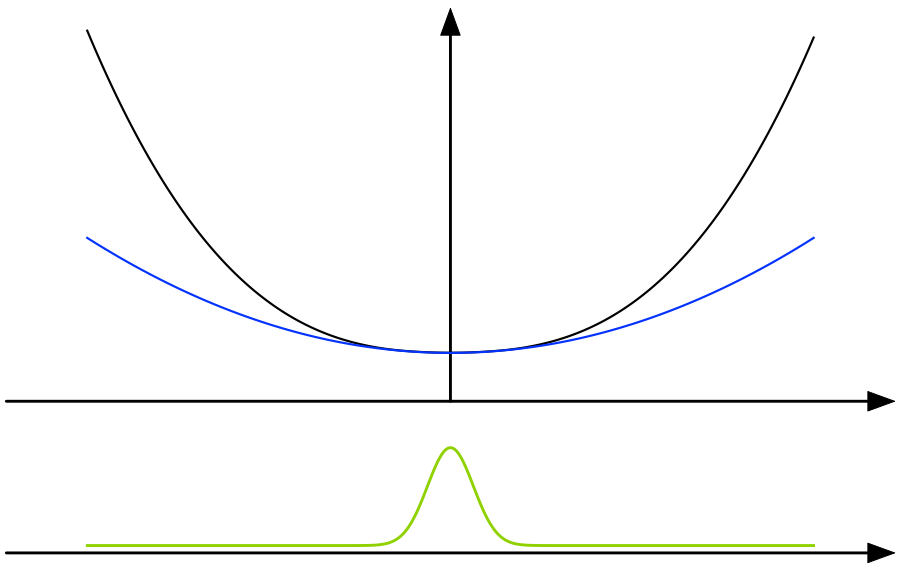


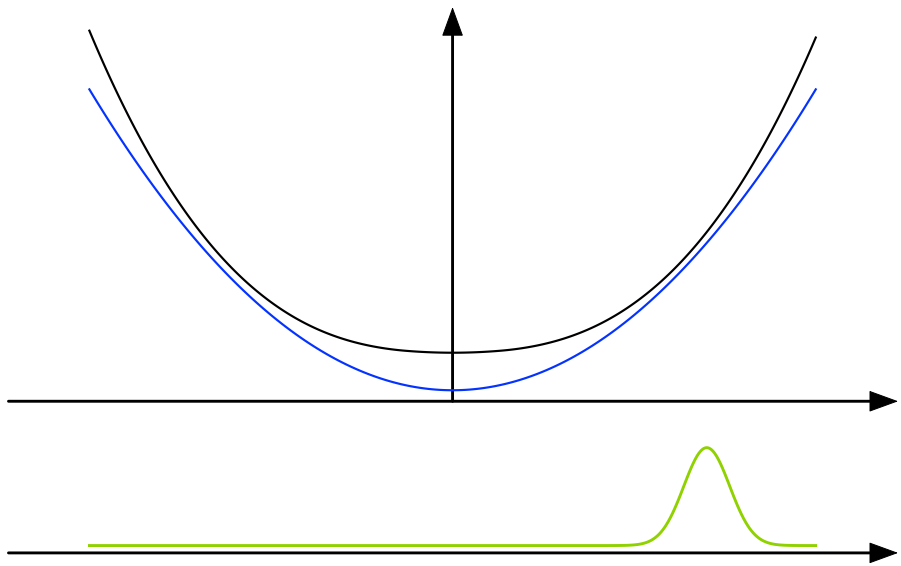
LP Approach to ADP



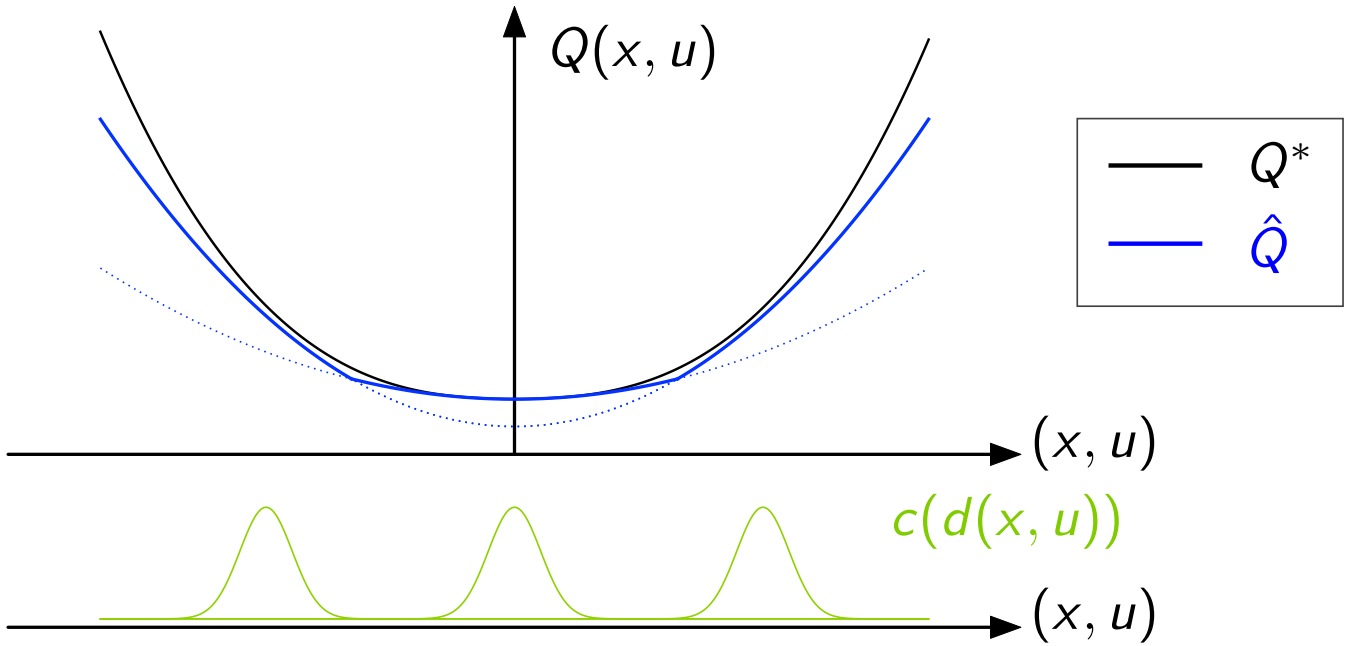
LP Approach to ADP

Which choice of tuning parameters is "best"?



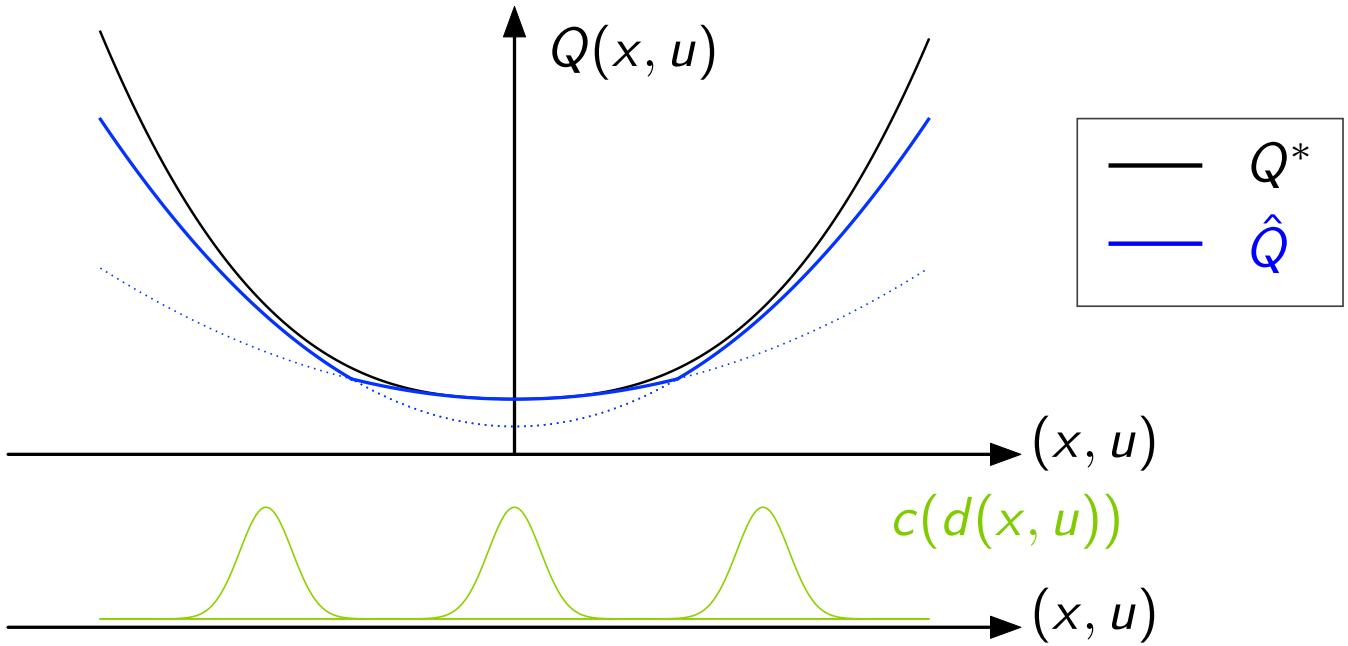


Point-wise Maximum of $\mathcal{Q}$-functions



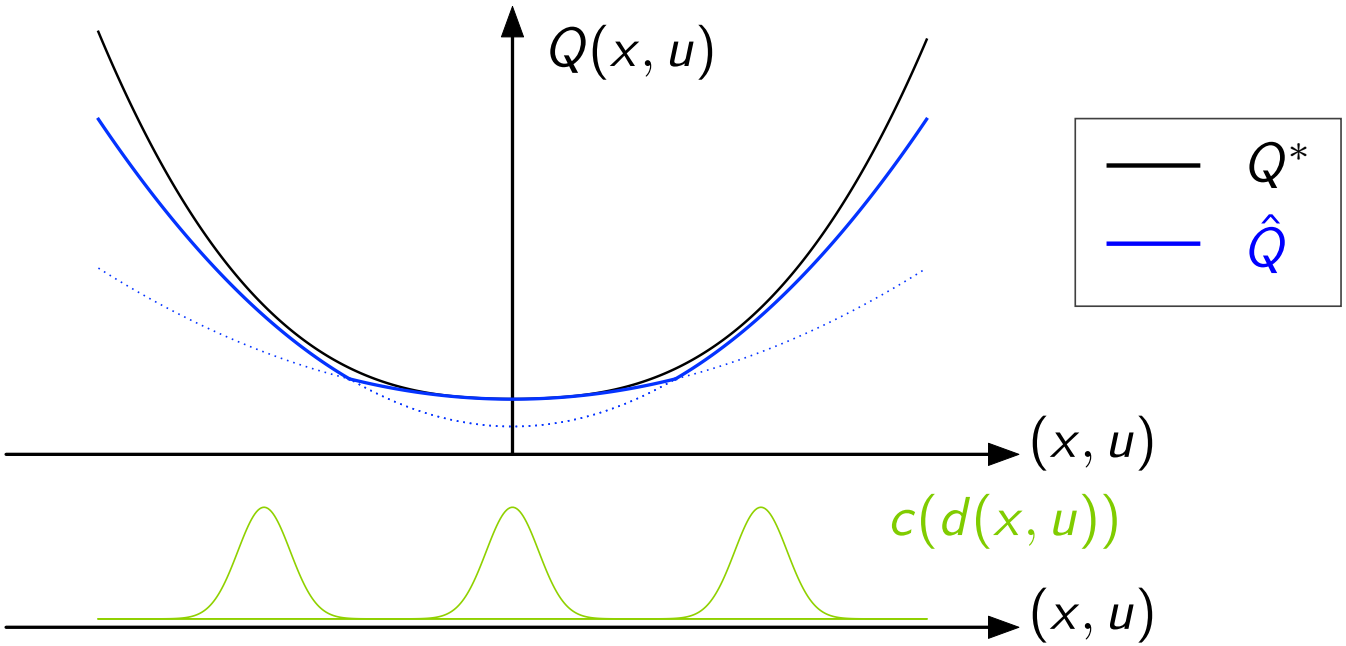
Point-wise Maximum of $\mathcal{Q}$-functions

1. Pick family: $\{c_j\}_{j\in\mathcal{J}}$, solve for family: $\left\{\hat{Q}_j\right\}_{j\in\mathcal{J}}$
2. Use point-wise maximum:



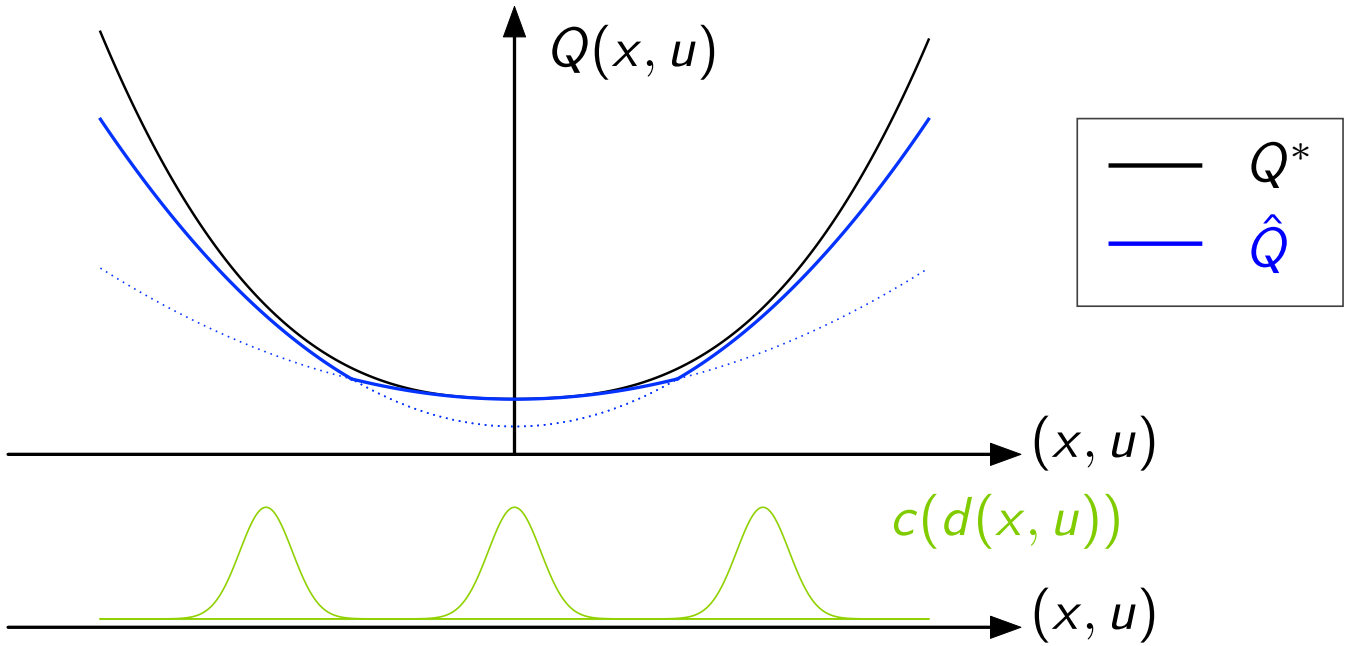
Point-wise Maximum of $\mathcal{Q}$-functions

1. Pick family: $\{c_j\}_{j\in\mathcal{J}}$, solve for family: $\{\hat{Q}_j\}_{j\in\mathcal{J}}$
2. Use point-wise maximum: $\hat{Q}(x,u) = \max_{j\in\mathcal{J}} \hat{Q}_j(x,u)$



Point-wise Maximum of $\mathcal{Q}$-functions

$$\text{Approx. Policy: } \hat{\pi}(x_t) = \arg\min_{u \in \mathcal{U}} \left(\max_{j \in \mathcal{J}} \hat{Q}_j(x_t, u) \right)$$



Encoding the objective

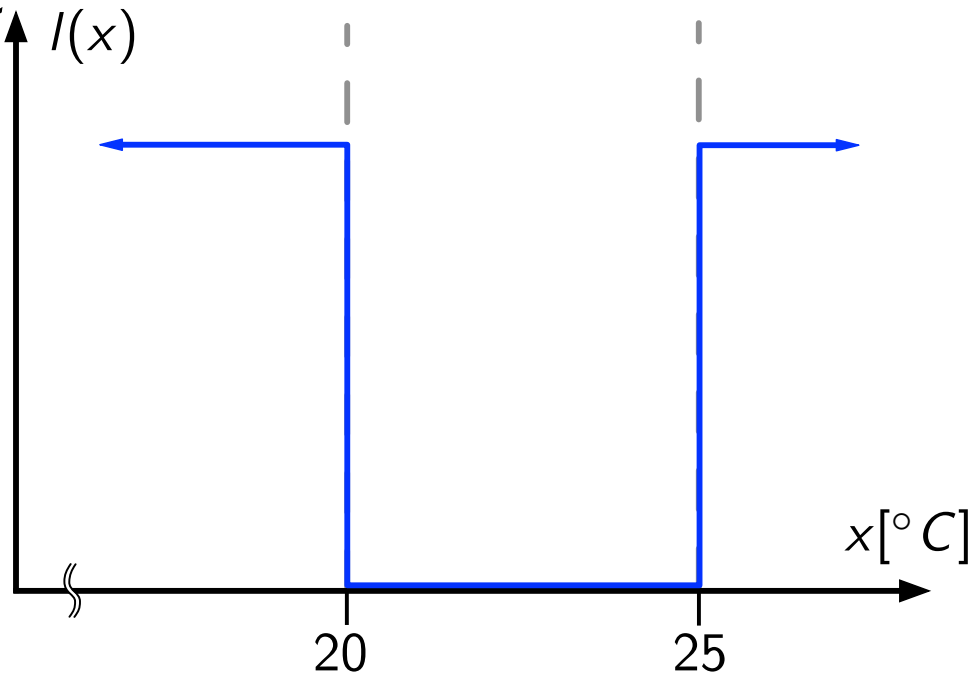
Objective: minimise energy consumption, while maintaining the room temperature within the desired comfort range of $20 - 25^{\circ}C$

Encoding the objective

Objective: minimise energy consumption, while maintaining the room temperature within the desired comfort range of $20 - 25^\circ C$

Use a penalty function:

- Ideal
- Quadratic

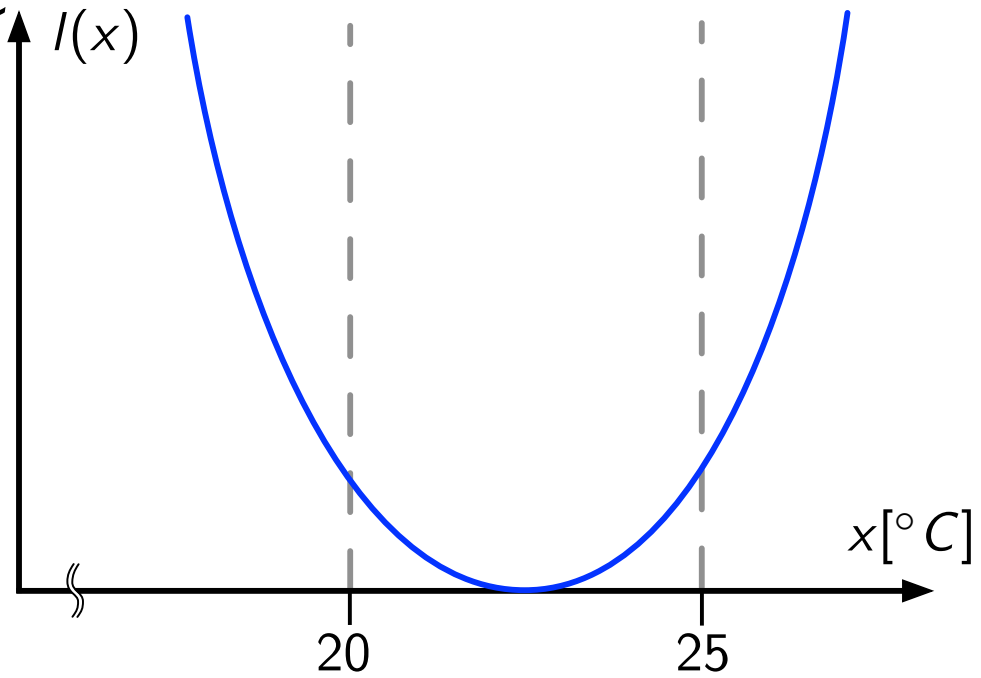


Encoding the objective

Objective: minimise energy consumption, while maintaining the room temperature within the desired comfort range of $20 - 25^\circ C$

Use a penalty function:

- Ideal
- Quadratic



Connection to PCAO

Both methods attempt to approximate the optimal Value function (i.e., V^* or Q^*)

Connection to PCAO

Both methods attempt to approximate the optimal Value function (i.e., V^* or Q^*)

ADP

$$\hat{Q}(x,u) = \max_{j \in \mathcal{J}} \hat{Q}_j(x,u)$$

Connection to PCAO

Both methods attempt to approximate the optimal Value function (i.e., V^* or Q^*)

ADP

$$\hat{Q}(x,u) = \max_{j \in \mathcal{J}} \hat{Q}_j(x,u)$$

P-CAO

$$\hat{V}(x) = \sum_{j \in \mathcal{J}} \beta_j(x)\, \hat{V}_j(x)$$

Connection to PCAO

Both methods attempt to approximate the optimal Value function (i.e., V^* or Q^*)

ADP

$$\hat{Q}(x,u) = \max_{j \in \mathcal{J}} \hat{Q}_j(x,u)$$

P-CAO

$$\hat{V}(x) = \sum_{j \in \mathcal{J}} \beta_j(x) \hat{V}_j(x)$$

Outlook

Next steps:

- Demonstrate the max of quadratics $\mathcal{Q}$-function on the more realistic building model
- Demonstrate the max of quadratics $\mathcal{Q}$-function on the real building

Open questions:

- A method for using ADP to warm start P-CAO
- Can the theorectical perforamcne guarantees from ADP be adapted to provide any thoeretical guarentees for P-CAO or other learning algrithms

Outlook

Next steps:

- Demonstrate the max of quadratics $\mathcal{Q}$-function on the more realistic building model
- Demonstrate the max of quadratics $\mathcal{Q}$-function on the real building

Open questions:

- A method for using ADP to warm start P-CAO
- Can the theorectical perforamcne guarantees from ADP be adapted to provide any thoeretical guarentees for P-CAO or other learning algrithms

Outlook

Next steps:

- Demonstrate the max of quadratics Q-function on the more realistic building model
- Demonstrate the max of quadratics Q-function on the real building

Open questions:

- A method for using ADP to warm start P-CAO
- Can the theorectical perforamcne guarantees from ADP be adapted to provide any thoeretical guarentees for P-CAO or other learning algrithms

Outlook

Next steps:

- Demonstrate the max of quadratics Q-function on the more realistic building model
- Demonstrate the max of quadratics Q-function on the real building

Open questions:

- A method for using ADP to warm start P-CAO
- Can the theorectical perforamcne guarantees from ADP be adapted to provide any thoeretical guarentees for P-CAO or other learning algrithms