**2021 IEEE Conference on Control Technology and Applications (CCTA).
August 8-11, 2021. San Diego, California**

**2021 IEEE Conference on Control Technology and Applications (CCTA).
August 8-11, 2021. San**

2021 IEEE Conference on Control Technology and Applications (CCTA) | 978-1-6654-3643-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/CCTA48906.2021.9658723

Model-Free Perception-Based Control via Q-Learning with an Application to Heat-Seeking Missile Guidance

Wade S. Kovalik, Lijing Zhai, Kyriakos G. Vamvoudakis

Abstract— **Modern perception-based sensing schemes incorporate machine learning and high-dimensional image observations to control system states, but face issues of perception error and incomplete dynamics and state information. To address these issues, we propose a novel perception-based control strategy using model-free output feedback Q-learning that incorporates a Faster R-CNN convolutional neural network. We specifically investigate the optimal control problem of a linear time-invariant, discrete-time system given only the observation image data. We evaluate the data-driven control design process in ideal perception and degraded perception conditions. We show that the resulting controller from output feedback Q-learning is non-optimal, but the optimality loss is bounded with bounded perception error. Simulated results on a simple missile, whose seeker head observes synthetic images of the target heat source modeled as a blurry ball of light, show the efficacy of the proposed model-free perception-based control framework.**

Index Terms— **Perception, output feedback, Q-learning, convolutional neural network.**

I. Introduction

Modern autonomous systems often incorporate sensing schemes that measure the environment in real time to control system states. For example, many quadcopters are equipped with laser rangefinders or LiDAR to determine their altitude or distance to obstacles. Particular attention is given to *perceptual* sensing schemes, where observations instead of sensor data contain information about system states. Observations are usually in the form of images, where confounding information adds difficulty in the form of perception error. This shift in focus is largely driven by the advent of self-driving cars, which incorporate camera suites to detect other vehicles, avoid pedestrians, and maintain lanes. This shift is also driven by the popularization of machine learning techniques in industry with real-time image stream processing [1]. However, perception-based control presents many issues not present in simpler sensor-based control schemes:

1) The observations are usually high-dimensional, with camera images often in the megapixel range [1], [2]. Thus, perception schemes must process potentially billions of bits of information, which makes real-time control exceedingly difficult.
2) The perception error, i.e., the difference between the true output and the perceived output from the observations, is highly nonlinear and state-dependent. Thus, it is difficult to design a robust controller that uses a perception-sensing modality [2], [3].
3) Perception-based control is usually implemented on board complex systems with highly nonlinear dynamics, such as automobiles and unmanned aerial vehicles [1]. This renders standard system identification techniques and linear control design insufficient. In addition, full or even partial knowledge of the system dynamics is often prohibitive to determine with such complex systems.
4) Rate information, such as velocity, is difficult to discern from still images observed by the perception scheme. Thus, accurate full-state feedback is often impossible to implement with perception-based control.

Machine learning techniques address the first issue of high-dimensional observations. Convolutional neural networks (CNNs) in particular are able to handle inputs of large images much more effectively than other neural network types [4], [5]. The perception error of such CNNs, however, depends on a variety of factors including training time, training data, and how confounding the environment is, making the perception error difficult to characterize.

To handle this perception error issue, recent research has produced theoretic robustness guarantees for perception-based control. The work of [6] trained a CNN to use color image observations to produce a sequence of intermediate states, which were used as targets for a model-based optimal controller implemented on robots. The authors of [2] utilized System Level Synthesis (SLS) to derive an additional robustness constraint for the controller synthesis, where uncertainties from perception-based sensors with tractable data-driven safety guarantees were quantified. The resulting robust controller drove the system trajectories close to the CNN training data set such that the perception error remained bounded. The work of [7] then applied this robust SLS controller to a quadrotor. Moreover, the authors in [8] proved the existence of a trade-off between accuracy and robustness in perception-based control.

However, all the aforementioned works assumed full knowledge of the system dynamics, and they implemented full-state feedback using techniques such as visual inertial odometry that introduced additional perception error as state estimation errors accumulated over time [7]. Therefore, it is desirable to implement both model-free and output feedback control for the perception-based scheme. The objective of this work is to address all four mentioned issues of perception-based control. Specifically, our work examines the feasibility of implementing a perception-based, output feed-

W. S. Kovalik, L. Zhai, and K. G. Vamvoudakis are with the Daniel Guggenheim School of Aerospace Engineering at the Georgia Institute of Technology, Atlanta, GA, USA, email: {wkovalik,zhailijing,kyriakos}@gatech.edu

This work was supported in part by the Department of Energy under grant No. DE-EE0008453 and NSF under grant Nos. CPS-1851588 and S&AS-1849198.

back reinforcement learning algorithm to generate control sequences. Reinforcement learning solves for optimal control policies online, forward in time, along state trajectories, and without knowing the system dynamics [9]–[11], which makes it promising for the perception-based control of autonomous systems.

Contributions: The contributions of our present work are threefold. First, we develop a model-free perception-based control framework. Second, we evaluate the data-driven control design process in ideal perception and degraded perception conditions. Finally, we apply our model-free perception-based control framework to a heat-seeking missile application.

Structure: The remainder of our paper is structured as follows. Section II formulates the perception-based optimal control problem. Section III then presents the data-driven perception and model-free output feedback control. Section IV presents a background on a simple missile, while Section V discusses the simulation results. The last section summarizes the entire work and talks about future directions.

II. Problem Formulation

Consider the linear time-invariant (LTI) system, $\forall k \in \mathbb{N}$,

$$x_{k+1} = Ax_k + Bu_k, \tag{1}$$

$$z_k = q(x_k), \tag{2}$$

where $x_k \in \mathbb{R}^n$ is the state, $u_k \in \mathbb{R}^m$ is the control action or input, $z_k \in \mathbb{R}^M$ is the high-dimensional observation, and $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are the dynamical matrices. The observation z_k represents raw images taken by an M-pixel camera according to the observation process $q(x_k)$. The system (1) has an output process, $\forall k$,

$$y_k = Cx_k, \tag{3}$$

where $y_k \in \mathbb{R}^p$ is the true output and $C \in \mathbb{R}^{p\times n}$ is the output matrix. We assume the system matrices (A, B, C) and observation map q are unknown; thus, the true output y_k cannot be directly computed from the state x_k. Instead, we define a perception process $p(z_k)$ which inputs the observation image z_k and produces the *perceived* output $\hat{y}_k \in \mathbb{R}^p$, $\forall k$,

$$\hat{y}_k = p(z_k) = y_k + e_k = Cx_k + e_k, \tag{4}$$

where $e_k \in \mathbb{R}^p$ is the perception error [2], [7]. The perception map p is learned and thus known; however, the perception error is as yet unknown since C is unknown.

We define *ideal perception* as the absence of perception error such that $\forall k\ e_k \equiv 0$ and $\hat{y}_k = y_k$. We define *degraded perception* as the presence of non-zero perception error such that $\exists k\ e_k \neq 0$ and $\hat{y}_k \neq y_k$, and so the perception process (4) can be thought of as a noisy sensor.

Suppose the control follows a fixed feedback policy μ, such that $\forall k\ u_k = \mu(\cdot)$. We aim to find the optimal policy that minimizes the infinite-horizon cost function, $\forall (x_k, e)$,

$$V^{\mu}(x_k, e) = \sum_{i=k}^{\infty} \gamma^{i-k}(\hat{y}_i^{\mathrm{T}} Q \hat{y}_i + u_i^{\mathrm{T}} R u_i), \tag{5}$$

where $Q = Q^{\mathrm{T}} \in \mathbb{R}^{p\times p} \geq 0$ and $R = R^{\mathrm{T}} \in \mathbb{R}^{m\times m} \succ 0$ are user-defined weighting matrices and $0 < \gamma \leqslant 1$ is a discount factor introduced to make the cost (5) finite and to mitigate biasing effects due to probing noise [12].

In the ideal perception case (usually $\gamma = 1$), minimizing (5) becomes the linear quadratic regulator (LQR) problem. Assuming (A, B, C) are known and full-state feedback is implemented, there exists a unique stabilizing optimal policy $\mu^\star$ such that, $\forall k$,

$$u_k^\star = \mu^\star(x_k) = -(R/\gamma + B^{\mathrm{T}} P^\star B)^{-1} B^{\mathrm{T}} P^\star A x_k, \tag{6}$$

where $P^\star = (P^\star)^{\mathrm{T}} \succ 0$ satisfies the Ricatti equation

$$\gamma[A^{\mathrm{T}}PA - A^{\mathrm{T}}PB(R/\gamma + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA] - P + C^{\mathrm{T}}QC = 0 \tag{7}$$

under the conditions of (A, B) being stabilizable and (A, O) being observable with $O^{\mathrm{T}}O = C^{\mathrm{T}}QC$.

The reinforcement learning techniques of [12], [13] instead find the optimal controller (6) in an online, model-free way using only input-true output data. This output feedback approach is of particular interest in our perception-based setting, since it is difficult to measure rate information from still images z_k. In the degraded perception case, however, the reinforcement learning techniques of [12], [13] are not guaranteed to produce the optimal controller (6). Our goal then is to implement the output-feedback reinforcement learning using perceived output data (4), then quantify the deleterious effects (if any) of non-zero perception error.

There are two main problems to be addressed: (1) how to learn a suitable perception map p such that the perception error is bounded; (2) how to find the optimal control policy in a model-free way. Our solution strategy will be to use a CNN to learn the perception map, and then use output feedback Q-learning to generate the control sequence based on the input and perceived output data.

III. Data-Driven Perception and Control

In this section, we present our perception-based output feedback Q-learning framework as summarized in Figure 1.

A. Data-Driven Perception via CNN

We utilize a convolutional neural network that accepts the observation images z_k as an input and produces the perceived outputs $\hat{y}_k$. Our missile simulation images will be in the kilopixel range. So compared to other neural network types, the convolution and pooling operations of a CNN allow the large images to be processed much faster and with less computational expense. For ease of implementation, we exploit transfer learning on a pre-trained deep CNN called Faster R-CNN [4], which is available from the PyTorch library for Python 3. Instead of initializing a new neural network with random weights and biases, transfer learning initializes the pre-trained Faster R-CNN, whose weights and biases have already been trained using an ImageNet data set. The weights and biases are then further tuned using our custom synthetic image data set such that the CNN is adapted for the missile simulation. With transfer learning implemented, the training time is drastically reduced since the weights and biases only need to be adjusted by small amounts.

We later demonstrate that the perception map learned from the Faster R-CNN produces a small and bounded perception error for the regulation of a moving ball. This feature is useful for the perception-based Q-learning discussed next.

B. Data-Driven Control via Output Feedback Q-Learning

Here we extend the output feedback Q-learning approach of [13] by applying it to our perception-based setting. We first define the ideal and degraded Q-functions, then make boundedness guarantees in the presence of perception error, and finally implement Q-learning to solve for the control policy in a model-free way. We point the reader to [10], [14] for more details on Q-learning.

Notation: Given any vector $\eta_k \in \mathbb{R}^l$, we define its *data vector* as $\bar{\eta}_{k-1,k-N} = \begin{bmatrix}\eta_{k-1}^{\mathrm{T}} & \eta_{k-2}^{\mathrm{T}} & \cdots & \eta_{k-N}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{lN}$.

1) Ideal Perception Q-Function: In the case of ideal perception, the following lemma from [12] allows us to represent the state x_k in terms of input-true output data.

Lemma 1. *Assume (A, C) is observable with an observability index K. Then the state can be represented as, $\forall k$,*

$$x_k = \begin{bmatrix} M_u & M_y \end{bmatrix} \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \end{bmatrix}, \tag{8}$$

where $M_u = U_N - A^N(V_N^{\mathrm{T}}V_N)^{-1}V_N^{\mathrm{T}}T_N \in \mathbb{R}^{n\times mN}$, $M_y = A^N(V_N^{\mathrm{T}}V_N)^{-1}V_N^{\mathrm{T}} \in \mathbb{R}^{n\times pN}$, and $N \geqslant K$, with matrices $U_N = \begin{bmatrix} B & AB & \cdots & A^{N-1}B \end{bmatrix} \in \mathbb{R}^{n\times mN}$, $V_N = \begin{bmatrix} (CA^{N-1})^{\mathrm{T}} & \cdots & (CA)^{\mathrm{T}} & C^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{pN\times n}$, and

$$T_N = \begin{bmatrix} 0 & CB & CAB & \cdots & CA^{N-2}B \\ 0 & 0 & CB & \cdots & CA^{N-3}B \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & CB \\ 0 & \cdots & \cdots & 0 & 0 \end{bmatrix} \in \mathbb{R}^{pN\times mN}.$$

We next define the *ideal Q-function* as the following recursive Bellman equation, given a policy μ and $\forall(x_k, u_k)$,

$$Q^\mu(x_k, u_k) = y_k^{\mathrm{T}}Qy_k + u_k^{\mathrm{T}}Ru_k + \gamma Q^\mu(x_{k+1}, \mu(x_{k+1})), \tag{9}$$

which is the cost of selecting an arbitrary control u_k at time k plus the cost of implementing a fixed state feedback policy $u_{k+1} = \mu(x_{k+1})$ from time $k+1$ onward. Note that $Q^\mu(x_{k+1}, \mu(x_{k+1})) = V^\mu(x_{k+1})$. Assuming that the cost is quadratic in state such that $V^\mu(x_k) = x_k^{\mathrm{T}}Px_k$, where $P = P^{\mathrm{T}} \succ 0$, then substitution of (8) for x_k and (1) for x_{k+1} into the ideal Q-function (9) yields, $\forall(x_k, u_k)$,

$$\begin{aligned} Q^\mu(x_k, u_k) &= \gamma\phi_k^{\mathrm{T}}S\phi_k \qquad (10) \\ &= \gamma \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \\ u_k \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} S_{\bar{u}\bar{u}} & S_{\bar{u}\bar{y}} & S_{\bar{u}u} \\ S_{\bar{y}\bar{u}} & S_{\bar{y}\bar{y}} & S_{\bar{y}u} \\ S_{u\bar{u}} & S_{u\bar{y}} & S_{uu} \end{bmatrix} \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \\ u_k \end{bmatrix}, \end{aligned}$$

where $\phi_k = \begin{bmatrix} \bar{u}_{k-1,k-N}^{\mathrm{T}} & \bar{y}_{k-1,k-N}^{\mathrm{T}} & u_k^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{mN+pN+m}$ is the input-true output data vector and $S = S^{\mathrm{T}} \in \mathbb{R}^{(mN+pN+m)\times(mN+pN+m)}$ is the Q-function kernel matrix.

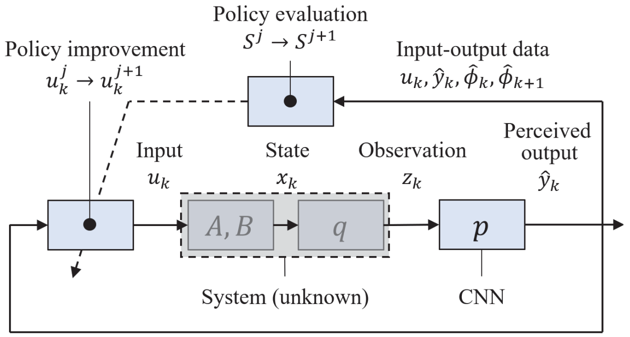

Fig. 1: Proposed perception-based output feedback framework using reinforcement learning and a convolutional neural network for model-free control.

The partitioned matrices in S are $S_{\bar{u}\bar{u}} = M_u^{\mathrm{T}}(C^{\mathrm{T}}QC/\gamma + A^{\mathrm{T}}PA)M_u \in \mathbb{R}^{mN\times mN}$, $S_{\bar{y}\bar{u}} = M_y^{\mathrm{T}}(C^{\mathrm{T}}QC/\gamma + A^{\mathrm{T}}PA)M_u = S_{\bar{u}\bar{y}}^{\mathrm{T}} \in \mathbb{R}^{pN\times mN}$, $S_{\bar{y}\bar{y}} = M_y^{\mathrm{T}}(C^{\mathrm{T}}QC/\gamma + A^{\mathrm{T}}PA)M_y \in \mathbb{R}^{pN\times pN}$, $S_{u\bar{u}} = B^{\mathrm{T}}PAM_u = S_{\bar{u}u}^{\mathrm{T}} \in \mathbb{R}^{m\times mN}$, $S_{u\bar{y}} = B^{\mathrm{T}}PAM_y = S_{\bar{y}u}^{\mathrm{T}} \in \mathbb{R}^{m\times pN}$, and $S_{uu} = R/\gamma + B^{\mathrm{T}}PB \in \mathbb{R}^{m\times m}$.

A necessary condition for optimality is the stationarity condition $\partial Q^\mu(x_k, u_k)/\partial u_k = 0$. Application to (10) yields the optimal output feedback policy $\mu^\star$ such that, $\forall k$,

$$\begin{aligned} u_k^\star &= \mu^\star(\bar{u}_{k-1,k-N}, \bar{y}_{k-1,k-N}) \\ &= -S_{uu}^{-1}(S_{u\bar{u}}\bar{u}_{k-1,k-N} + S_{u\bar{y}}\bar{y}_{k-1,k-N}). \end{aligned} \tag{11}$$

Substitution of (8) proves that (11) is equivalent to the state feedback controller (6) that solves the LQR problem [13].

2) Degraded Perception Q-Function: In the case of degraded perception, we define the *perceived state* $\hat{x}_k$ by replacing the true output data in (8) with the perceived output data, $\forall k$,

$$\hat{x}_k = \begin{bmatrix} M_u & M_y \end{bmatrix} \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{\hat{y}}_{k-1,k-N} \end{bmatrix} = x_k + M_y\bar{e}_{k-1,k-N}. \tag{12}$$

Note that $\bar{\hat{y}}_{k-1,k-N} = \bar{y}_{k-1,k-N} + \bar{e}_{k-1,k-N}$. We next define the *degraded Q-function*, $\forall(\hat{x}_k, u_k)$,

$$\begin{aligned} Q^\mu(\hat{x}_k, u_k) &= \gamma\hat{\phi}_k^{\mathrm{T}}S\hat{\phi}_k \qquad (13) \\ &= \gamma \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{\hat{y}}_{k-1,k-N} \\ u_k \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} S_{\bar{u}\bar{u}} & S_{\bar{u}\bar{y}} & S_{\bar{u}u} \\ S_{\bar{y}\bar{u}} & S_{\bar{y}\bar{y}} & S_{\bar{y}u} \\ S_{u\bar{u}} & S_{u\bar{y}} & S_{uu} \end{bmatrix} \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{\hat{y}}_{k-1,k-N} \\ u_k \end{bmatrix}, \end{aligned}$$

where the partitioned matrices in S are still defined as in the ideal Q-function (10). Application of stationarity $\partial Q^\mu(\hat{x}_k, u_k)/\partial u_k = 0$ yields the policy $\hat{\mu}^\star$ such that, $\forall k$,

$$\begin{aligned} \hat{u}_k^\star &= \hat{\mu}^\star(\bar{u}_{k-1,k-N}, \bar{\hat{y}}_{k-1,k-N}) \\ &= -S_{uu}^{-1}(S_{u\bar{u}}\bar{u}_{k-1,k-N} + S_{u\bar{y}}\bar{\hat{y}}_{k-1,k-N}). \end{aligned} \tag{14}$$

Note that $\hat{u}_k^\star$ is *not* necessarily equivalent to the optimal controller $u_k^\star$ from (6) due to the presence of potentially non-zero perception error in $\bar{\hat{y}}_{k-1,k-N}$.

For the degraded Q-function (13) to produce an admissible control policy, we must now guarantee that any deleterious effects of non-zero perception error are bounded.

Lemma 2. *Assume a degraded perception process* (4) *is implemented such that* $\forall k\ \|e_k\| \neq 0$. *Then the control policy in* (14) *does not minimize the cost* (5).

Proof. Replacement of $\bar{\hat{y}}_{k-1,k-N} = \bar{y}_{k-1,k-N} + \bar{e}_{k-1,k-N}$ into (14) yields the controller $\hat{u}^\star_k = -S^{-1}_{uu}(S_{u\bar{u}}\bar{u}_{k-1,k-N} + S_{u\bar{y}}\bar{y}_{k-1,k-N}) - S^{-1}_{uu}S_{u\bar{y}}\bar{e}_{k-1,k-N}$. Inspection with the optimal controller (11) implies that $\hat{u}^\star_k - u^\star_k = \Delta u^\star_k = -S^{-1}_{uu}S_{u\bar{y}}\bar{e}_{k-1,k-N}$. Since $\forall k\ \|e_k\| \neq 0$, $\|\Delta u^\star_k\| \neq 0$ and so the policy in (14) does not equal the optimal policy in (11) that minimizes the cost (5). This completes the proof. ■

Lemma 3. *Assume the perception error in Lemma 2 is bounded, i.e.,* $\forall k\ \exists M > 0$ *such that* $\|e_k\| \leqslant M$. *Then the control difference* $\Delta u^\star_k = \hat{u}^\star_k - u^\star_k$ *is also bounded, i.e.,* $\forall k\ \exists L > 0$ *such that* $\|\Delta u^\star_k\| \leqslant L$.

Proof. It follows from Lemma 2 and the definition of the induced norm that $\|\Delta u^\star_k\| = \|S^{-1}_{uu}S_{u\bar{y}}\bar{e}_{k-1,k-N}\| \leqslant \|S^{-1}_{uu}S_{u\bar{y}}\|\|\bar{e}_{k-1,k-N}\|$. This completes the proof. ■

We now define the *optimality loss* as the cost difference between implementing the non-optimal policy $\hat{\mu}^\star$ in the degraded perception case (14) and the optimal policy $\mu^\star$ in the ideal perception case (11) such that, $\forall k$,

$$\Delta V = V^{\hat{\mu}^\star}(x_k, e) - V^{\mu^\star}(x_k, 0). \tag{15}$$

The optimality loss serves as our metric for the deleterious effects of perception error. We prove the boundedness of the optimality loss in the following theorem.

Theorem 1. *Assume that* (A, B) *is stabilizable and* (A, O) *is observable. Assume the perception error is bounded, i.e.,* $\forall k\ \exists M > 0$ *such that* $\|e_k\| \leqslant M$. *Then the optimality loss* (15) *is also bounded, i.e.,* $\forall k\ \exists L > 0$ *such that* $|\Delta V| \leqslant L$.

Proof. Substituting $\hat{y}_k = y_k + e_k$ and $\hat{u}^\star_k = u^\star_k + \Delta u^\star_k$ into (15) and upon further expansion yields $\Delta V = \sum_{i=k}^{\infty} \gamma^{i-k}[e_i^{\mathrm{T}}Qe_i + 2e_i^{\mathrm{T}}Qy_i + (\Delta u^\star_i)^{\mathrm{T}}R\Delta u^\star_i + 2(\Delta u^\star_i)^{\mathrm{T}} Ru^\star_i]$. Using the Cauchy-Schwarz inequality, the first and third terms inside the summation can be bounded as $|e_k^{\mathrm{T}}Qe_k| \leqslant \rho(Q)\|e_k\|^2$ and $|(\Delta u^\star_k)^{\mathrm{T}}R\Delta u^\star_k| \leqslant \rho(R)\|\Delta u^\star_k\|^2$, respectively, where $\rho(\cdot)$ denotes the spectral radius and $\|\Delta u^\star_k\|$ is bounded from Lemma 3. The second and fourth terms can be bounded as $|e_k^{\mathrm{T}}Qy_k| \leqslant \|e_k\|\|Qy_k\|$ and $|(\Delta u^\star_k)^{\mathrm{T}}Ru^\star_k| \leqslant \|\Delta u^\star_k\|\|Ru^\star_k\|$, respectively. Since (A, B) is stabilizable and (A, O) is observable, the system (1) under the optimal policy in (6) is asymptotically stable, so $\|x_k\|$ is bounded and thus $\|y_k\|$ and $\|u^\star_k\|$ are also bounded. $|\Delta V|$ is now entirely bounded. This completes the proof. ■

Theorem 1 demonstrates that although the resulting controller (14) is no longer optimal in degraded perception case, the deleterious effects of non-zero perception error (in the form of the optimality loss) are bounded.

3) Degraded Perception Q-Learning: Currently, the Q-function kernel in (13) still requires knowledge of the system dynamics (A, B, C). The reinforcement learning strategy of Q-learning instead allows us to solve for the kernel in a model-free way.

The perception-based *policy iteration* (PI) algorithm of Q-learning is summarized in Algorithm 1. We select PI over other Q-learning algorithms (e.g., cost iteration [10], episodic Q-learning [15]) since PI evaluation is generally faster, which is helpful to pair with the Faster R-CNN perception model. Implementation of PI often involves parameterizing S and $\hat{\phi}_k$ in vector-basis form, then performing least-squares with probing noise to estimate the elements in S. We point the reader to [12], [13] for an exhaustive procedure.

Algorithm 1: Policy Iteration for Perception-based Output Feedback Q-Learning

Initialization. Given the degraded Q-function (13), begin with any kernel S^0 and stabilizing policy μ^0. Then for $j = 0, 1, 2, ...$, perform until convergence, $\|S^{j+1} - S^j\| < \varepsilon$:

Policy Evaluation. Using current policy, where $u^j_k = \mu^j(\cdot)$, solve for S^{j+1} such that

$$\hat{\phi}_k^{\mathrm{T}} S^{j+1} \hat{\phi}_k = \hat{y}_k^{\mathrm{T}} Q \hat{y}_k + (u^j_k)^{\mathrm{T}} R(u^j_k) + \gamma \hat{\phi}_{k+1}^{\mathrm{T}} S^{j+1} \hat{\phi}_{k+1}.$$

Policy Improvement. Update policy to μ^{j+1}, where

$$u^{j+1}_k = -(S^{j+1}_{uu})^{-1}(S^{j+1}_{u\bar{u}}\bar{u}_{k-1,k-N} + S^{j+1}_{u\bar{y}}\bar{\hat{y}}_{k-1,k-N}).$$

IV. Heat-Seeking Missile Background

We shall test our framework in a simulation of a heat-seeking missile, whose seeker head generates synthetic images of a moving blurry white ball representing the infrared exhaust signature of a target aircraft. The missile setup is shown in Figure 2. We first derive a state-space representation for the angle of attack and sideslip angle of the missile. Then, we implement a simple line-of-sight guidance law where the missile controller attempts to regulate the position of the ball to the center of the image. The intent is to keep the line of sight constant, thus ensuring intercept with the target [16]. Finally, we use the Faster R-CNN for the perception process and Q-learning to solve for the controller in the presence of degraded perception. We compare any optimality loss to an LQR controller of ideal perception.

A. Linearized Missile Dynamics

Assumption 1. Assume an axisymmetric cruciform missile body; no roll dynamics; negligible gravity differential effects; the relative velocity vector of the missile always pointing along the line of sight, i.e., toward the ball; and the missile always being sufficiently far away from the target. □

Define the missile state to be $x = [\alpha\ \dot{\alpha}\ \beta\ \dot{\beta}]^{\mathrm{T}}$, where α is the angle of attack and β is the sideslip angle. The control input is $u = [\delta_e\ \delta_r]^{\mathrm{T}}$, where δ_e is the elevator fin deflection and δ_r is the rudder fin deflection. With the assumption of no roll dynamics, elevator deflection only effects a change in angle of attack and angle-of-attack rate, while rudder deflection only effects a change in sideslip angle and sideslip rate. By starting with the exact nonlinear six-degree-of-freedom equations of motion for aircraft dynamics [17], then linearizing about the steady level flight conditions $\bar{x} = [0\ 0\ 0\ 0]^{\mathrm{T}}$

and $\bar{u} = [0\ 0]^{\mathrm{T}}$ and applying Assumption 1, we get LTI, continuous-time second-order dynamics where the angle-of-attack and sideslip-angle dynamics are decoupled—i.e., $x = [\alpha\ \dot{\alpha}]^{\mathrm{T}}$ and $u = \delta_e$ for the angle-of-attack system and $x = [\beta\ \dot{\beta}]^{\mathrm{T}}$ and $u = \delta_r$ for the sideslip-angle system. The dynamics matrices A and B are pulled from [18]. The angle-of-attack system has the dynamics, $\forall t \geqslant 0$,

$$\begin{bmatrix} \dot{\alpha} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} -2.7 & 1 \\ -5.5 & -0.4 \end{bmatrix} \begin{bmatrix} \alpha \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0.27 \\ -19 \end{bmatrix} \delta_e. \tag{16}$$

A zero-order hold with a sampling time of 0.01 s is applied to discretize (16), $\forall k \in \mathbb{N}$,

$$\begin{bmatrix} \alpha_{k+1} \\ \dot{\alpha}_{k+1} \end{bmatrix} = \begin{bmatrix} 0.97 & 0.0098 \\ -0.054 & 0.99 \end{bmatrix} \begin{bmatrix} \alpha_k \\ \dot{\alpha}_k \end{bmatrix} + \begin{bmatrix} 0.17 \\ -19 \end{bmatrix} \delta_{e,k}. \tag{17}$$

Since the missile is axisymmetric, the discretized linearized sideslip dynamics are identical to (17) with the corresponding state $x_k = [\beta_k\ \dot{\beta}_k]^{\mathrm{T}}$ and control input $u_k = \delta_{r,k}$.

B. Observation Process using Synthetic Images

To simulate the missile seeker head, synthetic images z_k are generated given the current missile state x_k. The images are of a moving blurry white ball on a black background similar to [2]. The images are 240 × 240 pixels in size for computation and storage considerations. A white 30-pixel diameter ball is first drawn on the image such that the entire ball is inside the image boundary, which is then blurred using a Gaussian filter. Given Assumption 1, the sideslip angle and angle of attack are directly proportional to the x- and y-coordinates of the center of the ball on the image, respectively. The true output is thus $y_k = [X_k\ Y_k]^{\mathrm{T}}$, where the pair (X, Y) represents the coordinates of the center of the ball in pixels. The gain between α and Y (which depends on the target distance, the pixel pitch of the camera sensor, and the focal length of the camera lens) is set to be 10° or 0.175 rad per 120 pixels. Thus, the angle of attack is limited to $\pm 10^\circ$ which allows the linearized system (17) to accurately describe the missile dynamics, and the output process is $Y_k = [-687.5\ 0]\,[\alpha_k\ \dot{\alpha}_k]^{\mathrm{T}}$. The same gain is chosen between β and X such that the sideslip output process is identical.

C. Perception Process using Trained Faster R-CNN

Given a synthetic image, the Faster R-CNN outputs the coordinates of a bounding box around the ball such that the perceived output $\hat{y}_k = [\hat{X}_k\ \hat{Y}_k]^{\mathrm{T}}$ is simply the midpoint of the bounding box. Note the bounding box has pixel resolution, and so the midpoint will have half-pixel resolution; this contributes to the perception error. The transfer learning process that learns the perception map p to generate these bounding boxes is as follows: First, a set of 440 images is created with the ball at random locations on the image. The data set is then randomly batched into a 400-image training data set and a 40-image validation data set. The Faster R-CNN is trained using the training data set, whereas the validation data set is used to assess the accuracy at the end of every training epoch. The Faster R-CNN is trained for five epochs. The perception error is assessed by passing 1000 images with random ball locations through Faster R-CNN. Average accuracy of the perceived output is 99.8% with a minimum accuracy of 98.2% (i.e., perception error magnitude of 0.25 and 3 pixels, respectively). Such low errors after only five epochs of training demonstrates the advantage of transfer learning. More importantly, it demonstrates that the perception error is bounded.

D. Missile Control Policy via Q-Learning

Given the state x_k, input u_k, and perceived output $\hat{y}_k$, we define Q-functions for the angle-of-attack system and for the sideslip-angle system in the form of (13). The angle-of-attack system (17) is of second order, so we select $N = 2$ for the data vectors. If the Q-function is instead parameterized in vector-basis form, where the redundant elements in S are removed, then Q-learning requires at least 15 samples to fully solve for the elements of S via least-squares (thus updating every 0.15 s). The same procedure is applied simultaneously to the sideslip-angle system such that the sequences for $\delta_{e,k}$ and $\delta_{r,k}$ are generated in parallel.

V. Simulation Results

We wish to regulate the ball to the center of the image, i.e., to $y = [0\ 0]^{\mathrm{T}}$. This is a pure pursuit guidance law and will result in intercept if the missile-to-target speed ratio does not exceed two [16]. If lead pursuit is desired, then the ball can simply be regulated to another point in the image given the angle-off is known. We pick $Q = 1$, $R = 1$, and $\gamma = 0.5$. The PI algorithm can then be initialized with $S^0 = I$ and $u_k^0 = 0$, which is stabilizing since the system (17) is open-loop stable. For each iteration step, input-perceived output data are collected over 15 time steps to estimate the Q-function kernel using least-squares with zero-mean Gaussian probing noise added to the control to maintain excitation. The PI algorithm is then iterated until below $\varepsilon = 0.025$.

Figure 3 shows the normalized perceived output trajectories of the angle-of-attack system for eight random initial ball locations, along with the optimal LQR trajectory as reference. We observe that the Q-learning control sequences are always able to regulate the ball to the center of the image and hold its position thereafter, and so the missile is successfully guided until intercept. Moreover, the missile is guided without knowing any angle-of-attack rate information nor the system dynamics, showing our perception-based Q-learning framework successfully learns the control policy in a model-free way. Figure 2 shows a trajectory animation as observed by the missile seeker head for one of the runs. Here, the control sequence for both the angle of attack and sideslip angle are found simultaneously. The Faster R-CNN produces very little perception error throughout.

Figure 3 also shows the effects of non-zero perception error. For all eight runs, the degraded Q-function kernels $\hat{S}$ converge to values slightly different than the optimal kernel $S^\star$ (obtained via (10)), and so the generated control sequences are indeed non-optimal. In addition, the Q-learning control sequences regulate the ball much slower than

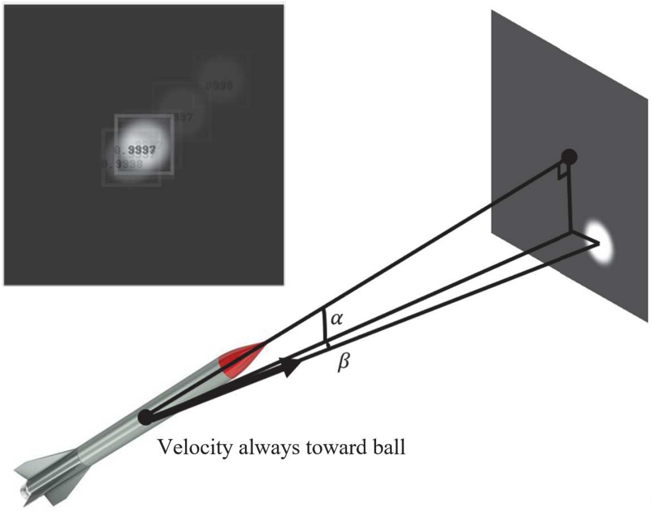

Fig. 2: Missile perception setup. (inset) Animation for an initial location of $y_0 = [55\ 65]^{\mathrm{T}}$ pixels. Still frames taken every 20 time steps up to $k = 200$, with accuracy numbers and progressively lighter frames as the animation progresses.

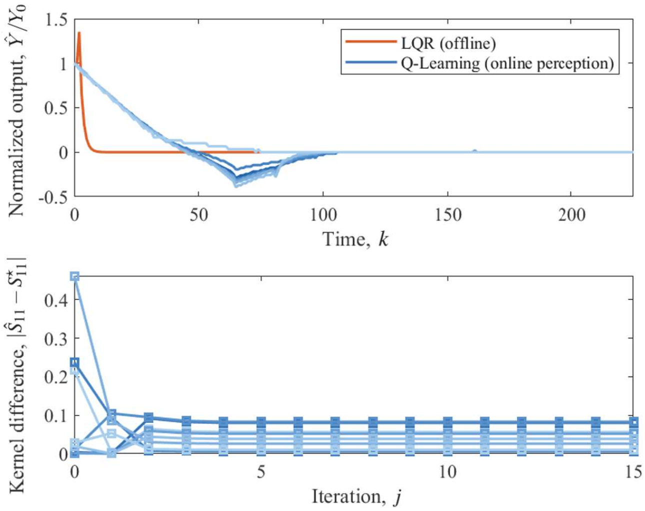

Fig. 3: Normalized output trajectories and Q-function kernel convergence (first entry plotted) of the angle-of-attack system for eight random initial ball locations. LQR trajectory included as reference.

the LQR sequence, likely due to perception error and the 0.15-s kernel update time. However, Figure 3 also validates Theorem 1 as the kernel differences (and thus optimality losses) remain bounded throughout the simulation.

VI. Conclusions and Future Work

In this work, we proposed a perception-based framework by incorporating a Faster R-CNN neural network into the control policy design via output feedback Q-learning. We evaluated the data-driven control design process in the degraded perception condition. We demonstrated that the optimality loss of output feedback Q-learning with bounded perception error was bounded. Simulation results on a simple missile showed the efficacy of the proposed model-free perception-based control framework.

Future work directions include relaxing many of the missile assumptions by introducing roll dynamics and gravity dynamics; considering more realistic environments with confounding heat signatures (e.g., the sun, ground clutter, defensive flares); and characterizing the perception error using a data-driven model, then incorporating it into the Q-learning framework to reduce the optimality loss.

References

[1] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[2] S. Dean, N. Matni, B. Recht, and V. Ye, "Robust guarantees for perception-based control," in *Learning for Dynamics and Control*. PMLR, 2020, pp. 350–360.

[3] Y. Abbasi-Yadkori and C. Szepesvári, "Regret bounds for the adaptive control of linear quadratic systems," in *Proceedings of the 24th Annual Conference on Learning Theory*. JMLR Workshop and Conference Proceedings, 2011, pp. 1–26.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[5] B. T. Nugraha, S.-F. Su *et al.*, "Towards self-driving car using convolutional neural network and road lane detector," in *2017 2nd International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT)*. IEEE, 2017, pp. 65–69.

[6] S. Bansal, V. Tolani, S. Gupta, J. Malik, and C. Tomlin, "Combining optimal control and learning for visual navigation in novel environments," in *Conference on Robot Learning*. PMLR, 2020, pp. 420–429.

[7] L. Jarin-Lipschitz, R. Li, T. Nguyen, V. Kumar, and N. Matni, "Robust, perception based control with quadrotors," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2020, pp. 7737–7743.

[8] A. A. Al Makdah, V. Katewa, and F. Pasqualetti, "Accuracy prevents robustness in perception-based control," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 3940–3946.

[9] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal control*. John Wiley & Sons, 2012.

[10] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Systems Magazine*, vol. 32, no. 6, pp. 76–105, 2012.

[11] D. Vrabie, K. G. Vamvoudakis, and F. L. Lewis, *Optimal adaptive control and differential games by reinforcement learning principles*. IET, 2013, vol. 2.

[12] F. L. Lewis and K. G. Vamvoudakis, "Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 1, pp. 14–25, 2010.

[13] S. A. A. Rizvi and Z. Lin, "Output feedback reinforcement q-learning control for the discrete-time linear quadratic regulator problem," in *2017 IEEE 56th annual conference on decision and control (CDC)*. IEEE, 2017, pp. 1311–1316.

[14] T. Landelius and H. Knutsson, "Greedy adaptive critics for lqr problems: Convergence proofs," Linköping, Sweden, October 1996.

[15] C. Watkins, "Learning from delayed rewards," Ph.D. dissertation, Cambridge University, Cambridge, U.K., 1989.

[16] F. P. Adler, "Missile guidance by three-dimensional proportional navigation," *Journal of Applied Physics*, vol. 27, no. 5, pp. 500–507, 1956.

[17] B. Etkin and L. Reid, *Dynamics of flight: Stability and control*, 3rd ed. Toronto, Canada: John Wiley & Sons, Inc., 1996.

[18] S. Das and K. Halder, "Missile attitude control via a hybrid lqg-ltr-lqi control scheme with optimum weight selection," in *2014 First International Conference on Automation, Control, Energy and Systems (ACES)*. IEEE, 2014, pp. 1–6.