Model-based and Model-free Reinforcement Learning for Visual Servoing

Amir-massoud Farahmand, Azad Shademan, Martin Jagersand, and Csaba Szepesvari
Uncalibrated Visual Servoing

Model Estimation

Locally Linear Regression

Learning/Planning Control Signal

Regularized Fitted Q-Iteration

Model-based and Model-free settings

Learner/Planner (e.g. RFQI)

Model Estimator (e.g. LLR)

objective function (reinforcement signal)

\( X^* \)

\( \{ S_i, \hat{X}_i \} \)
- Uncalibrated Visual Servoing
- Model Estimation
  - Locally Linear Regression
  - Learning/Planning Control Signal
  - Regularized Fitted Q-Iteration
  - Model-based and Model-free settings

Objective function (reinforcement signal)
Visual Servoing

Visual-servoing is the task of minimizing a visually-specified objective by giving appropriate control commands to a robot.

**Image Features**
- Geometric primitives
  - points
  - lines
- Patterns (region)

**Various Tasks**
- Point to Point
- Point to Line
- ...

...
\[ S \in \mathcal{R}^d \]
\[ X := [x_1; x_2; \ldots; x_m]_{m \times 1} \]
\[ F := [f_1(S); f_2(S); \ldots; f_m(S)]_{m \times 1} \]
\[ X = F(S). \]

\[ \frac{dX}{dt} = \frac{\partial F(S)}{\partial S} \frac{dS}{dt} = J(S) \frac{dS}{dt} \]

Jacobian
<table>
<thead>
<tr>
<th>Calibrated</th>
<th>Uncalibrated</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F(S)$ is known.</td>
<td>$F(S)$ is unknown.</td>
</tr>
</tbody>
</table>

Tedious or not feasible in many cases (e.g. after grasping an object, change in camera position, etc.)

$F(S)$ (or $J(S)$) is estimated.

Usually local estimation (RLS, Broyden, etc.)
Local vs. Global Visual Servoing

Local information (Jacobian) to design control signal

\[ e = X - X^* \Rightarrow \dot{e} = \frac{\partial F(S)}{\partial S} \dot{S} = J(S)e \]

Global [kinematic] information to design optimal control signal

\[ \dot{S} = Ke \Rightarrow \dot{e} = J(S)Ke \]

Choose \( K = -\lambda J(S)\dagger \Rightarrow \dot{e} = -J(S)J(S)\dagger e = -\lambda e \quad (\lambda > 0) \]

Better performance

Need \textbf{global} information about the system
Main Idea

- Use regression to estimate the **global** visual-motor kinematic system

  - Our choice: **Locally Linear Regression**

- Use reinforcement learning to learn/plan

  - Our choice: **Regularized Fitted Q-Iteration**
Uncalibrated Visual Servoing

Model Estimation
- Locally Linear Regression
- Learning/Planning Control Signal
- Regularized Fitted Q-Iteration
- Model-based and Model-free settings

Objective function (reinforcement signal)

Learner/Planner (e.g. RFQI)

Model Estimator (e.g. LLR)

$X^*$

$\{S_i, \hat{X}_i\}$
Locally Linear Regression for Visual-Motor Kinematic Modeling

Goal: Find a good estimate $\hat{F}(S)$ of $F(S)$

$$a_0^i(S), a_1^i(S) \leftarrow \arg \min_{a_0, a_1} \sum_{l=1}^{t} w_l(S) \left( x_i(l) - (a_0^i + a_1^i \cdot (S_l - S)) \right)^2$$

$$w_l(S) = K_m((S - S_l)/h) = \exp \left( \frac{\|S - S_l\|^2}{2h^2} \right) \text{ (example)}$$
Locally Linear Regression: Bandwidth Selection

- Bandwidth is important
- Model Selection: Cross-Validation (CV)
- CV for temporally-dependent data?
- A special k-fold CV method is introduced

$$K_m((S - S_l)/h) = \exp\left(\frac{\|S - S_l\|^2}{2h^2}\right) \text{ (example)}$$
Locally Linear Regression for Visual-Motor Kinematic Modeling

![Graph showing root mean squared error vs. model estimation sample size for two bandwidth settings: fixed (0.1) and CV. The graph indicates a decrease in error with increasing sample size for both settings, with the CV setting generally having a lower error.]
- Uncalibrated Visual Servoing
- Model Estimation
  - Locally Linear Regression
- Learning/Planning Control Signal
  - Regularized Fitted Q-Iteration
  - Model-based and Model-free settings

![Diagram](image)

Objective function (reinforcement signal)

\[ X^* \rightarrow \text{Learner/Planner (e.g. RFQI)} \]

\( \{ S_i, \hat{X}_i \} \)

\[ \pi(S') \rightarrow \text{Model Estimator (e.g. LLR)} \]

Control signal generator
RL for Control Signal Design

- Design global controller
- No model is needed
  - Use data
- Two modes of operation
  - Model-free
  - Model-based
Model-free vs. Model-based Reinforcement Learning

Model-Free RL (Learning)
Use samples $(S,X)$ directly coming from the robot

Model-Based RL (Planning)
Use virtual samples coming from the estimated visual-servoing kinematic model
Value Iteration

Find $V$ (or $Q$) such that $Q = T^*Q$ (Bellman optimality equation)

$$(TQ)(x, a) = r(x, a) + \gamma \max_{a' \in A} \int Q(y, a') P(dy | x, a).$$
Fitted Value Iteration

\[ Q_{k+1} = \arg \min_{Q \in \mathcal{F}^M} \frac{1}{M_k} \sum_{i=1}^{M_k} \left[ Q(X_i, A_i) - \left( R_i + \gamma \max_{a' \in A} Q_k(X'_i, a') \right) \right]^2 \]

\[ Q_{k+1} \leftarrow \hat{T}^* Q_k \]

No access to \( T^* Q_k \). Just \( \hat{T}^* Q_k \) (samples)
The Choice of Function Space?

- Dependence on data
- Parametric approach is limited
Non-Parametric Approach and Ill-Posed Optimizations

$$Q_{k+1} = \arg \min_{Q \in \mathcal{F}^M} \frac{1}{M_k} \sum_{i=1}^{M_k} \left[ Q(X_i, A_i) - \left( R_i + \gamma \max_{a' \in A} Q_k(X'_i, a') \right) \right]^2$$

Remove [most of the] assumptions on the function space
Regularization for Model-Selection

\[ Q_{k+1} = \arg \min_{Q \in \mathcal{F}} \frac{1}{M_k} \sum_{i=1}^{M_k} \left[ Q(X_i, A_i) - \left( R_i + \gamma \max_{a' \in \mathcal{A}} Q_k(X'_i, a') \right) \right]^2 + \lambda \text{Pen}(Q) \]

1. Remove [most of the] assumptions on the function space
2. Find the right region of that space using regularization
RKHS-based Formulation

Reproducing Kernel Hilbert Spaces
Defined by a Kernel

Gaussian Processes interpretation

\[ Q_{k+1} = \arg \min_{Q \in \mathcal{H}^M} \frac{1}{M_k} \sum_{i=1}^{M_k} \left[ Q(X_i, A_i) - \left( R_i + \gamma \max_{a' \in \mathcal{A}} Q_k(X_i', a') \right) \right]^2 + \lambda \| Q \|^2_H \]

\[ Q(x, a) = \sum_{i=1}^{M_k} \alpha_i k((X_i, A_i), (x, a)) \quad \text{Two parameters: } \lambda \text{ and kernel parameter} \]

\[ \alpha^{(k+1)} = (K + M_k \lambda I)^{-1} (r + \gamma K^+ \alpha^k) \]

Closed-Form Solution
**Theorem 1 (L²-bound):** Assume that \( S = [0, 1]^d \), \( k \in \text{Lip}^*(\alpha, C(S, S)) \), \( \alpha > d \), and \( Q_k \) is such that \( TQ_k \in \mathcal{H} (= \mathcal{H}_k) \).\(^4\) Furthermore, (for the sake of simplicity) assume that all functions involved in the regression problem (the reward function, \( Q_k \), and the result of the optimization problem \( Q_{k+1} \)) are bounded by some constant \( L > 0 \).\(^5\) Let \( Q_{k+1} \) be the solution of (4) with some \( \lambda > 0 \). Furthermore, assume that we use the same number of samples in each iteration: \( M_1 = M_2 = \ldots = M_K \). Let \( \pi_K \) be greedy w.r.t. the \( K^{th} \) iterate, \( Q_K \). Define \( E_0 = ||\varepsilon - 1||_\infty \) and let \( B = \max_{0 \leq k \leq K} ||T_k^kQ_0||^2_\mathcal{H} \). Then, for any \( \delta > 0 \) with probability at least \( 1 - \delta \),

\[
\|V^* - V^{\pi_K}\|_\rho \leq \\
2 \left[ \frac{1}{1 - \gamma} + \frac{\gamma}{(1 - \gamma)^2} \right]^{\gamma K/2} E_0 + \\
2 \left[ \frac{(C_{\rho, \nu})^{1/2}}{1 - \gamma} + \gamma (C_{\rho, \nu})^{1/2} \right] \times \\
\left[ c_1 \lambda B + \frac{c_2 L^4}{M_1^{d/\alpha}} + \frac{c_3 \log(1/\delta)}{M_1 L^4} \right]^{1/2}
\]

for some constants \( C_{\rho, \nu}^1 \) and \( C_{\rho, \nu}^2 \) that only depend on \( \rho, \nu, \gamma \) and the MDP dynamics and for some universal constants \( c_1, c_2, c_3 > 0 \).
Uncalibrated Visual Servoing
- Model Estimation
  - Locally Linear Regression
- Learning/Planning Control Signal
  - Regularized Fitted Q-Iteration
  - Model-based and Model-free settings

Learner/Planner (e.g. RFQI)

Model Estimator (e.g. LLR)

\( X^* \)

\( \{ S_i, \hat{X}_i \} \)

Objective function (reinforcement signal)

\( \pi(S') \)

Control signal generator
Experimental Setup

- Puma robot
- Stereo cameras
- 3 Degrees of Freedom
- 4-dim feature space
- 2 actions/joint $\Rightarrow$ 8 actions
- Reward: $-1$ every step except when it is close to the goal ($X^*$)
Visual Servoing without Model Selection

Sample size vs. Average return for different control methods:
- Model-free RL
- Model-based RL
- Linear controller

MBRL uses twice virtual samples.
Effect of Regularization Coefficient and Kernel Parameter
Visual Servoing
with Model Selection

---

Average return vs. Sample size

- Linear controller
- Model-free RL (with model selection)
- Model-free RL (without model selection)
Summary

Objective function (reinforcement signal)

\[ X^* \]

\( \{ S_i, \hat{X}_i \} \)

Learner/Planner (e.g. RFQI)

\( \pi(S') \)

Model Estimator (e.g. LLR)

Control signal generator

Control

Model

Estimator

(e.g. LLR)

Objective function (reinforcement signal)

Control

Model

Estimator

(e.g. LLR)

Objective function (reinforcement signal)

Control

Model

Estimator

(e.g. LLR)

Objective function (reinforcement signal)

Control

Model

Estimator

(e.g. LLR)

Objective function (reinforcement signal)

Control

Model

Estimator

(e.g. LLR)

Objective function (reinforcement signal)
Conclusion

- Both model-free and model-based RL (with RFQI) worked well for visual-servoing task.
- Model-based RL performed better (twice virtual sample size).
- Model selection is important.
  - Model Estimation
  - Learning/Planning
Future Work

- Real robot, e.g. WAM
- Other robotics applications (any sequential decision-making problem)
- Efficient model selection?
  - use estimated model?
  - cross-validation at each iteration?
- When model-based RL would outperform model-free RL?
  - Learning the dynamics is easier than learning the action-value function
- Continuous actions
Thank You!
Locally Linear Regression for Visual-Motor Kinematic Modeling

$$\hat{x}_i(S) = \sum_{l=1}^{t} b_l(S) x_i(l)$$

where $b(S)^T = (b_1(S), \cdots, b_t)$, and $b(S)^T = e_1^T(S_s W_s S_s)^{-1} S_s W_s$, $e_1 = (1, 0, \cdots, 0)^T$, and

$$S_s = \begin{pmatrix}
1 & S_1 - S \\
1 & S_2 - S \\
\vdots & \vdots \\
1 & S_t - S
\end{pmatrix}$$

$$w_l(S) = K_m((S - S_l)/h) = \exp\left(\frac{\|S - S_l\|^2}{2h^2}\right)$$ (example)
Locally Linear Regression for Visual-Motor Kinematic Modeling

Goal: Find a good estimate $\hat{F}(S)$ of $F(S)$

\[
\begin{aligned}
a^i_0(S), a^i_1(S) &\leftarrow \arg \min_{a_0, a_1} \sum_{l=1}^t w_l(S) \left( x_i(l) - (a^i_0 + a^i_1 \cdot (S_l - S)) \right)^2 \\
w_l(S) &= K_m((S - S_l)/h) = \exp \left( \frac{\|S - S_l\|^2}{2h^2} \right) \text{ (example)}
\end{aligned}
\]

\[
\begin{aligned}
\hat{F}(S) &= [\hat{x}_1(S); \cdots; \hat{x}_m(S)] \\
\hat{x}_i(S) &= \sum_{l=1}^t b_l(S)x_i(l) \\
b(S)^T &= (b_1(S), \cdots, b_t(S)) = e_1^T(S_s^T W_s S_s)^{-1} S_s^T W_s
\end{aligned}
\]

\[
\begin{aligned}
S_s &= \begin{pmatrix} 1 & S_1 - S \\ 1 & S_2 - S \\ \vdots & \vdots \\ 1 & S_t - S \end{pmatrix} \\
W_s[i, i] &= K_m((S_i - S)/h)
\end{aligned}
\]
Cross-Validation for LLR Model Selection

Conventional k-fold CV for i.i.d. processes

k-fold CV for mixing processes