Lecture 9

Gradient-Free Trajectory Optimization

All of the trajectory optimization approaches we have explored have required gradient information. Recently, given the advent of massively parallel processing, methods that leverage sampling have led to a number of gradient-free trajectory optimization approaches.

9.1. Path Integral Control

Path-integral control attempts to solve the stochastic trajectory optimization problem for an uncertain system.

Consider the stochastic differential equation

$$dX_t = f(X_t, t)dt + dW_t \tag{9.1}$$

where W_t is a wiener process.

The distribution $p(x_{k+1}, t_{k+1}|x_k, t_k)$ could be very complex.

Assume zero mean, where $\mathbb{E}[dW_t^T dW_t] = v(t, x, u)dt$.

As is typical, make the cost to be minimized an expectation given as

$$C(t, x, u) = \mathbb{E}\left[\Phi(X_T) + \int_t^T d\tau R(t, X_t, u(X_t, t))\right]. \tag{9.2}$$

Can we minimize this over all stochastic trajectories to optimize $u(X_t, t)$.

Let's once again apply the Bellman recursion:

$$J(t, X_t) = \min_{u_t}\left[R(t, x_t, u_t)dt + \mathbb{E}\left[J(t + dt, X_{t+dt})\right]\right]. \tag{9.3}$$

$$J(t + dt, x_t + dX_t) = J(t, X_t) + \frac{\partial J}{\partial t}dt + \frac{\partial J}{\partial x}dX_t + \nabla_{xt}J dX_t dt + \nabla_{tt}J dt^2 + \frac{1}{2}dX_t^T \nabla_{xx}J dX_t \tag{9.4}$$

We know that

$$\mathbb{E}[dX_t] = f(x_t, t)dt \tag{9.5}$$

and

$$
\begin{aligned}
dX_t^T \nabla_{xx} J dX_t &= (f dt + dW)^T \nabla_{xx} J (f dt + dW_t) && (9.6)\\
&= (f dt + dW_t)^T (\nabla_{xx} J f dt + \nabla_{xx} J dW_t) && (9.7)\\
&= f^T dt (\nabla_{xx} J f dt + \nabla_{xx} J dW_t) + dW_t^T (\nabla_{xx} J f dt + \nabla_{xx} J dW_t) && (9.8)\\
&= f^T dt \nabla_{xx} J f dt + f^T dt \nabla_{xx} J dW_t + dW_t^T \nabla_{xx} J f dt + dW_t^T \nabla_{xx} J dW_t && (9.9)
\end{aligned}
$$

We can then apply the cyclic property of the trace, such that $tr(dW_t^T \nabla_{xx} J dW_t) = tr(dW_t^T dW_t \nabla_{xx} J)$ and we have

$$
\mathbb{E}[dX_t^T \nabla_{xx} J dX_t] = f^T \nabla_{xx} J f dt^2 + \frac{1}{2} tr(v \nabla_{xx} J) dt. \tag{9.10}
$$

This gives

$$
\mathbb{E}\Big[J(t + dt, x + dX_t)\Big] = J(t, X_t) + \frac{\partial J}{\partial t} dt + \frac{\partial J}{\partial x} f dt + \frac{1}{2} tr(v \nabla_{xx} J) dt. \tag{9.11}
$$

We can then write

$$
-\frac{\partial J}{\partial t} = \min_u \Big[R(t, x, u) + f(x, u, t) \frac{\partial J}{\partial x} + \frac{1}{2} v \nabla_{xx} J\Big], \quad J(x, T) = \Phi(x) \tag{9.12}
$$

which is known as the stochastic HJB equation. We can solve this for linear systems, but not for arbitrary systems. Consider a slightly more complex system

$$
dX_t = f(x_t, t) dt + g(x, t)\Big[u(X_t, t) dt + dW_t\Big] \tag{9.13}
$$

and a cost

$$
C(t, x, u) = \mathbb{E}\Big[\Phi(x_T) + \int_t^T V(X_s, s) + \frac{1}{2} u^T(X_s, s) R u(X_s, s) ds\Big]. \tag{9.14}
$$

Here $g = \begin{bmatrix} 0_{l \times m} \\ g_c \end{bmatrix}$, where $u \in \mathbb{R}^m$, and $x \in \mathbb{R}^n$, and $l = n - m$. We can then write

$$
-\frac{\partial J}{\partial t} = \min_u \left(\frac{1}{2} u^T R u + V + \frac{\partial J}{\partial x}^T (f + gu) + \frac{1}{2} tr\Big(g v g^T \nabla_{xx} J\Big)\right) \tag{9.15}
$$

$$
J(x, t_f) = \Phi(x). \tag{9.16}
$$

This gives $u = -R^{-1} g^T \nabla J$. Substituting back into the HJB we get

$$
-\frac{\partial}{\partial t} = \frac{1}{2} \nabla_x J^T g R^{-1} g^T \nabla_x J + V + \nabla_x J^T f + \frac{1}{2} Tr(g v g^T \nabla_{xx} J). \tag{9.17}
$$

Let us define $J(x, t) = -\lambda \log \Psi(x, t)$ and $R = \lambda v^{-1}$, where $\lambda > 0$. After a lengthy derivation, which is omitted here, we can then write

$$
-\frac{\partial \Psi}{\partial t} = \left(-\frac{V}{\lambda} + f^T \nabla_x + \frac{1}{2} Tr\Big(g v g^T \nabla_{xx}\Big)\right)\Psi \tag{9.18}
$$

where $\Psi(x, T) = \exp(-\Phi(x)/\lambda)$. We can now apply the Feynman-Kac formula. Let $P(\tau | x, t)$ be a distribution over uncontrolled trajectories. That is where

$$
dX_t = f(X_t, t) dt + g(X, t) dW_t \tag{9.19}
$$

We can now solve for $\Psi(x,t)$ using the Feynman-Kac formula to get

$$\Psi(x,t) = \int dP(\tau|x,t)e^{-s(\tau)/\lambda} \tag{9.20}$$

where

$$s(\tau) = \Phi(x(T)) + \int_t^T V(x(s),s)ds \tag{9.21}$$

and

$$\Psi(x_{t_0},t_0) = \mathbb{E}_p\left[e^{-s(\tau)/\lambda}\right]. \tag{9.22}$$

To get u^* we take the gradient of Ψ with respect to x. If g_c is square, after another lengthy derivation, this yields

$$u^*dt = g_c(x_{t_0},t_0)^{-1}\frac{\mathbb{E}_p[\exp(-s(\tau)/\lambda)g_c(x_{t_0},t_0)dW]}{\mathbb{E}_p[\exp(-s(\tau)/\lambda)]}. \tag{9.23}$$

where c denotes the controlled portion of the dynamics. We can write this in discrete time as

$$dX_t = (f(X_t,t) + g(X_t,t)u(X_t,t))dt + g(X_t,t)\epsilon\sqrt{dt} \tag{9.24}$$

where ϵ is a time-varying vector of standard Gaussian random variables. The uncontrolled dynamics are

$$dX_t = f(X_t,t)dt + g(X_t,t)\epsilon\sqrt{dt} \tag{9.25}$$

and we can write

$$g(X_t,t)dW \approx dX_t - f(X_t,t)dt. \tag{9.26}$$

Given that

$$S(\tau) \approx \Phi(x_T) + \sum_{i=0}^{N} V(x_t,t)dt \tag{9.27}$$

we have

$$u^*dt = g_c(x_{t_0},t_0)^{-1}\frac{\mathbb{E}_p[\exp(-s(\tau)/\lambda)(dX_t^c - f^c(X_t,t)dt)]}{\mathbb{E}_p[\exp(-s(\tau)/\lambda)]}. \tag{9.28}$$

We can then write

$$u^* = g_c(x_{t_0},t_0)^{-1}\frac{\mathbb{E}_p[\exp(-s(\tau)/\lambda)(dX_t^c/dt - f^c(X_t,t))]}{\mathbb{E}_p[\exp(-s(\tau)/\lambda)]}. \tag{9.29}$$

However, sampling from the uncontrolled dynamics is very inefficient. Ideally, we would like to control the variance and mean of the distribution via importance sampling. Rewrite u^* by defining the expectation in integral form:

$$u^* = g_c(x_{t_0},t_0)^{-1}\frac{\int \exp(-s(\tau)/\lambda)(dX_t^c/dt - f^c(X_t,t))p(\tau)d\tau}{\int \exp(-s(\tau)/\lambda)p(\tau)d\tau}. \tag{9.30}$$

If you multiply through by $q(\tau)$, the distribution of the controlled dynamics, you have

$$u^* = g_c(x_{t_0},t_0)^{-1}\frac{\mathbb{E}_q[\exp(-s(\tau)/\lambda)(dX_t^c/dt - f^c(X_t,t))\frac{p(\tau)}{q(\tau)}]}{\mathbb{E}_q[\exp(-s(\tau)/\lambda)\frac{p(\tau)}{q(\tau)}]}. \tag{9.31}$$

where now the costs can be sampled from the distribution for the controlled dynamics. Here, $\frac{p(\tau)}{q(\tau)}$ is known as the likelihood ratio. How do you find $\frac{p(\tau)}{q(\tau)}$? Approximate the one-step dynamics as being sampled from Gaussian probability distribution, and the resulting likelihood ratio can be absorbed into the cost function, where $s(\tau)$ now becomes $\tilde{s}(\tau)$, and input u can also be extracted. For the special case of the dynamics

$$dX_t = f(X_t)dt + g(X_t, t)(u(X_t, t) + \frac{1}{\sqrt{\rho}}\frac{\epsilon}{\sqrt{dt}})dt \tag{9.32}$$

we can write

$$\delta u = \frac{1}{\sqrt{\rho}}\frac{\epsilon}{\sqrt{dt}} \tag{9.33}$$

and our update law becomes

$$u^* = u + \frac{\mathbb{E}_q[\exp(-\tilde{s}(\tau)/\lambda)\delta u]}{\mathbb{E}_q[\exp(-\tilde{s}(\tau)/\lambda)]} \tag{9.34}$$

where $\tilde{s}(\tau) = s(\tau) + \frac{1-\nu^{-1}}{2}\delta u^T R \delta u + u^T R \delta u + \frac{1}{2}u^T R u$. Here ν is the variance of the sampling distribution.

9.2. Cross-Entropy Method

Assume we have a performance metric H and we want to estimate

$$\ell = \mathbb{E}_p[H(Z)]. \tag{9.35}$$

We know that

$$\ell = \mathbb{E}_p[H(Z)] = \int H(z)p(z)dz \tag{9.36}$$

Assume another probability density q exists. Then we can write

$$\ell = \mathbb{E}_p[H(Z)] = \int H(z)p(z)\frac{q(z)}{q(z)}dz = \mathbb{E}_q\left[H(z)\frac{p(z)}{q(z)}\right] \tag{9.37}$$

where $q(z)$ is the importance density. To pick a good importance density, we want to minimize the variance of $\hat{\ell}$, where

$$\hat{\ell} = \frac{1}{N}\sum_{i=1}^{N} H(Z_i)\frac{p(Z_i)}{q(Z_i)} \tag{9.38}$$

so

$$q^* = \arg\min \mathbb{V}_q\left(H(Z)\frac{p(Z)}{q(Z)}\right) = \frac{H(z)p(z)}{\ell}, \tag{9.39}$$

since $\mathbb{V}_q(\ell) = 0$. However, since we are trying to estimate ℓ, there is no way to implement this. So, researchers use the Kullback-Leibler (KL) divergence as a distance between distributions p and q.

$$KL(p, q) = \int p(z)\log p(z)dz - \int p(z)\log q(z)dz \tag{9.40}$$

Assume p is parameterized by $\bar{v}$ such that $p(z, \bar{v})$. We often want to find a q that is as close as possible to this distribution, so we say $q = p(z, v)$. To find v we minimize

$$v^* = \arg\min_v KL(q^*, p(z, v)) \tag{9.41}$$

or

$$v^* = \max_v \int q^*(z) \log p(z, v) dz \tag{9.42}$$

$$= \max_v \int \frac{H(z)p(z, \bar{v})}{\ell} \log p(z, v) dz \tag{9.43}$$

$$= \max_v \int \frac{H(z)p(z, \bar{v})}{\ell} \log p(z, v) dz \tag{9.44}$$

$$= \max_v \mathbb{E}_{\bar{v}} \Big[H(Z) \log p(Z, v) \Big] \tag{9.45}$$

Now we consider the task of estimating a low probability event, namely that the cost is less than some small value.

$$\ell = \mathbb{P}_{\bar{v}}(J(Z) \leq \gamma) = \mathbb{E}_{\bar{v}}[\mathbb{I}_{J(Z) \leq \gamma}] \tag{9.46}$$

To determine the optimal v for this condition, we can use

$$v^* = \max_v \mathbb{E}_{\bar{v}} \Big[\mathbb{I}_{J(Z) \leq \gamma} \log p(Z, v) \Big]. \tag{9.47}$$

This can be approximated as

$$\hat{v}^* = \arg\max_v \frac{1}{N} \sum_{i=0}^{N} \mathbb{I}_{J(Z_i) \leq \gamma} \log p(Z_i, v) \tag{9.48}$$

However, if $J(Z) \leq \gamma$ is a rare event, this approximation will not be useful. The cross-entropy approach executes a multi-level approach. The algorithm is as follows:

1. Sample $Z_1 \dots Z_n$ from $p(z, \hat{v}_{i-1})$ and compute the ρ^{th} quantile $\hat{\gamma}$.
2. Update $\hat{v}_i$: $\hat{v}^* = \arg\max_v \frac{1}{N} \sum_{i=0}^{N} \mathbb{I}_{J(Z_i) \leq \hat{\gamma}} \log p(Z_i, v)$

Bibliography

[1] Marin Kobilarov. Cross-entropy motion planning. *The International Journal of Robotics Research*, 31(7):855–871, 2012.

[2] Grady Williams, Andrew Aldrich, and Evangelos A Theodorou. Model predictive path integral control: From theory to parallel computation. *Journal of Guidance, Control, and Dynamics*, 40(2):344–357, 2017.