Path Integral Control Theory

Team 4 **Natali Alfonso, Simon Kern, Vangelis Kostas**
Machine Learning (NWI-NM048C)
Prof. B. Kappen Due date: 31/01/2017

1 Solve the mass on the spring problem discussed in the tutorial on slides 32 to 35, 40 such that the end velocity is maximal.

The problem x and t are similar to the one in the slides. So we use the same reasoning to arrive to:

$$F = -z + u = \ddot{z} \tag{1}$$
$$x_1 = z \tag{2}$$
$$x_2 = \dot{z} \tag{3}$$

Combining (1) with (2) and (3) we get:

$$\dot{x_1} = x_2$$
$$\dot{x_2} = -x_1 + u$$

Since $R(x, u, t) = 0$ HJB in this case will take the same form as before:

$$-\frac{\partial J}{\partial t} = \min_{t \leq T-1} (x_2 \frac{\partial J}{\partial x_1} - x_1 \frac{\partial J}{\partial x_2} + u \frac{\partial J}{\partial x_2})$$

Which would reach a minimum for $u = -sign(\frac{\partial J}{\partial x2})$ (4) taking into account that $u \epsilon [-1, 1]$.

$$-\frac{\partial J}{\partial t} = x_2 \frac{\partial J}{\partial x_1} - x_1 \frac{\partial J}{\partial x_2} - \left| \frac{\partial J}{\partial x_2} \right|$$

Contrary to the case in the slides though in our case the border condition would be $\phi(x) = -x_2$ so minimizing that would give us the maximum x_2 and in turn maximum $\dot{z}$ which is the speed. Again we will try $J(t, x) = \psi_1(x_1) + \psi_2(x_2) + \alpha(t)$. For this form of J HJBE reduces to:

$$\dot{\psi_1} = \psi_2$$
$$\dot{\psi_2} = -\psi_1$$
$$\dot{\alpha} = -|\psi_2|$$

Our boundary conditions in this case will be $\psi_1(T) = 0$,$\psi_2(T) = -1$ and $\alpha(T) = 0$ So we have the solution:

$$\psi_1(t) = sin(t - T)$$
$$\psi_2(t) = -cos(t - T)$$

The optimal control for this J and taking into account expression (4) should be:

$$u(x,t) = -sign(\frac{\partial J}{\partial x_2}) = -sign(\psi_2(t)) = -sign(-cos(t - T))$$

2 Consider the control problem

2.1 Solve the control problem in the deterministic case $\nu = 0$ using the PMP formalism.

In this case we wont have noise so we start by writing the PMP equations:

$$\dot{\lambda} = -H_x(t, x(t), u(t), \lambda(t)) \tag{4}$$
$$0 = H_u(t, x(t), u(t), \lambda(t)) \tag{5}$$
$$\dot{x} = H_\lambda(t, x(t), u(t), \lambda(t)) \tag{6}$$
$$\lambda(T) = -\phi_x(x(T)) \tag{7}$$

By looking at the problem we can see that we have:

$$\phi(x(T)) = \frac{1}{2}x(T)^2$$
$$dx = udt \Rightarrow \dot{x} = u \Rightarrow f(x,u,t) = u$$
$$C = 0.5x(T)^2 + \int_{t_0}^{T} dt 0.5u(t)^2$$

So we can get the expressions we need to calculate the solution to the PMP:

$$R = \frac{1}{2}u(t)^2$$
$$f = u$$
$$\phi(x(T)) = \frac{1}{2}x(T)^2$$
$$-H = R - \lambda f = \frac{1}{2}u(t)^2 - \lambda(t)u(t)$$

By solving (5) for u we get $-\frac{1}{2}2u(t) + \lambda(t) = 0$ so $u^\star(t) = \lambda(t)$. By replacing this in the Hamiltonian we get $H^\star = -\frac{1}{2}\lambda(t)^2 + \lambda(t)^2 = \frac{1}{2}\lambda(t)^2$.
By using (4) and (6) we get:

$$\dot{x} = \lambda(t) \tag{8}$$
$$\dot{\lambda} = 0 \tag{9}$$

And finaly by solving these 2 differential equations we get:

$$\begin{aligned} x(t) &= A + Bt \\ \lambda(t) &= B \\ (7) \Rightarrow \lambda(T) &= -x(T) = B \\ B &= \frac{-A}{1+T} \end{aligned}$$

Where A and B are constants that will be solved to fit the boundary conditions when T and t_0, x_0 become known. A in this case would be $x(t_0)$, if we write the time in reference to t_0 which is not 0, and B can be replaced with λ so we get:

$$\begin{aligned} \lambda = x(T) &= x(t_0) + \lambda(T - t_0) \\ \Rightarrow u(x,t) = \lambda &= \frac{-x(t_0)}{1+T-t_0} \end{aligned}$$

2.2 Solve the control problem in this stochastic case using the Bellman equation

We notice that the equations are simplified in our case since most factors become zero(A=C=D=β=0) and others take scalar values(B=1,m=1,$\psi = \alpha$).So In this stochastic case with linear dynamics and quadratic cost, the HJB equation decouples into the following differential equations.

$$-\dot{P} = P^2 \tag{10}$$

$$\dot{\alpha} = P\alpha \tag{11}$$

$$\dot{\beta} = \frac{1}{2}\alpha^2 - \frac{1}{2}\nu P \tag{12}$$

And in the boundary we have $P(T) = 1$ and $\alpha(T) = 0$
We solve (10) for $P(t)$:

$$\begin{aligned} \alpha(t) &= 0 \\ P(t) &= \frac{1}{c-t} \\ P(T) = \frac{1}{c-T} &= 1 \Rightarrow c = 1+T \end{aligned}$$

From which we can find the optimal control:

$$u(x,t) = -P(t)x - \alpha(t) = -x\frac{1}{1+T-t}$$

2.3 Solve the control problem in the stochastic case using the path integral control methods and the Fokker Planck equation.

In this linear quadratic case the Fokker-Planck equation takes the form:

$$\dot{\rho} = -\frac{V}{\lambda}\rho - \nabla^T(0 * \rho) + \frac{1}{2}Tr(\nabla^2(gvg^T\rho))$$
$$\Rightarrow \rho(y, T|x, t) = \frac{1}{\sqrt{2\pi\sigma}}exp(-\frac{(y-x)^2}{2\sigma^2})$$

Which is a Gaussian with variance $\sigma^2 = \nu(T-t)$. Since we assumed that $J(x,t) = -\lambda log(\psi(x,t))$ we need to calculate ψ to get the cost-to-go.

$$\psi(x,t) = \int dy\rho(y, T|x, t)exp(-\phi(y)/\lambda)$$
$$\phi(x) = \frac{1}{2}x^2 \Rightarrow J(x,t) = \nu Rlog(\frac{\sigma}{\sigma_1} + \frac{1}{2}\frac{\sigma_1^2}{\sigma^2}x^2)$$

where $1/\sigma_1^2 = 1/\sigma^2 + \alpha/\nu R$. And from the cost-to-go we can get the optimal control:

$$u = -R^{-1}g^T\nabla J \Rightarrow u(x,t) = -R^{-1}\frac{\partial J}{\partial x} = -\frac{x}{R+T-t}$$

Which is the same result we got in the previous question for $R = 1$.

3 Consider the controlled random walk in one dimension.

3.1 Give an expression for the solution $\rho(y, t|x, 0)$. Show that the solution satisfies the Fokker Planck equation

$$\frac{1}{2}\nu\frac{\partial\rho}{\partial y}^2 = \frac{\mathrm{e}^{-\frac{(x-y)^2}{\sqrt{\nu\,(T-t)}}}\,(2\,x-2\,y)^2}{16\,\pi\,\nu\,(T-t)^2}$$

4 Write a Matlab program for the control problem in exercise 3

```
n=200;
J=[];
lambda=1;
for x=-2:0.1:2
    for v=-2:0.1:2
        E=0;
        E_all=[];
        for sample=1:n % perform Metropolis Hasting step
            E_new=FinalEnergy(x,v); % calculate the -phi(T)/lambda
            Delta=-(E_new-E);
            if (Delta < 0)
                E=E_new;
            else
                if (rand< exp(-Delta))
                    E=E_new;
                end;
            end;
            E_all=[E_all E];
        end;
        J=[J; [x,v,-lambda*log(mean(E_all))]] % append the new datapoint for J
    end;
end;
```

4.1 By varying ν, T, study numerically how the optimal control depends on these parameters

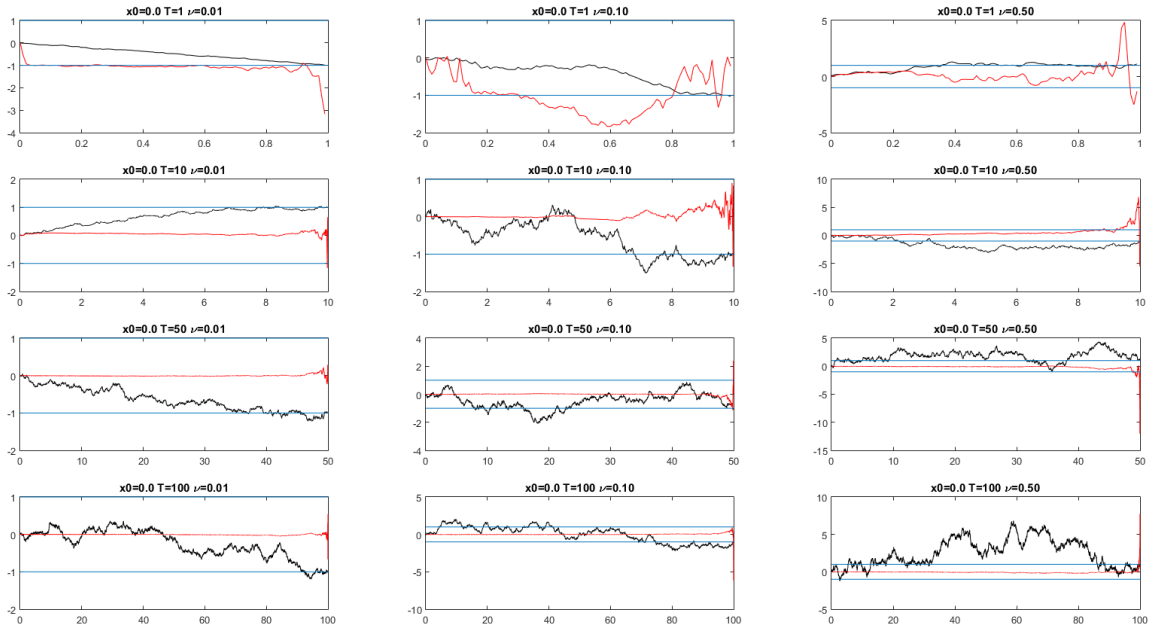

Figure 1: x in time.

In Figure 1 the red line represents the $u^\star$ control for each step and the black line is the x in that particular step. The time discretization we used was $dt = 0.01$. We have 2 horizontal blue lines in +/-1 to represent the target $\phi(x(T))$. The x always starts at 0.

As we can see from Figure 1 in the first column the noise is minimal and even in far targets T=100 we dont have a big fluctuation in the X during the steps.
As the noise grows in column 2 and 3 we start to see fluctuations in x which sometimes take it out of the target strip which raise the path cost. From what we can see the target of +/-1 is almost always reached at time T. It only becomes harder to do for larger T and bigger noise as we can see in the last row of column 3.
For the control $u^\star$ we notice that it tends to have big fluctuations as the target time T comes closer. This is because of the factor (T-t) in the numerator. These bigger values of $u^\star$ will tend to pull x towards its +/-1 target value more forcefully the closer it gets to the time T. For longer time periods the control seems to be close to 0 for most of the time letting x explore but it reacts if x changes too violently.

4.2 Explain in words the delayed choice mechanism.

We can see the delayed choice taking place in Figure 1. When T=1 the red line which is $u^\star$ takes a value far from 0 very close to time t_0. When T= 10,50,100 we see that if the noise is not overwhelmingly high the control will avoid making a choice(keeping close to 0) until time T is very close and it diverges from 0 having made a clear choice as to where x should be heading. This has to do with the (T-t) factor in the numerators.

5 Consider the mountain car problem

```
x_0=0.5;
v_0=0;
d_t=0.01;
nu=0.1;
T=10;
R=1;
g=9;

v=v_0;
x=x_0;
X_ALL=[];
u=0;%no optimal control
for t=0:d_t:T-d_t
    dl=dL(x);
    F=-g*dl/sqrt(1+dl^2);
    dv=F*d_t+u*d_t+normrnd(0,sqrt(nu*d_t)); %calculating dv with normal noise
    v=v+dv;
    dx=v*d_t;
    x=x+dx;
    X_ALL=[X_ALL x];% calculate and append the new x in time t
    if max(X_ALL)>2 || min(X_ALL)<-2
        if max(X_ALL)>2
            mm=max(X_ALL)
        else
            mm=min(X_ALL)
        end
        disp (sprintf('Made it out at x=%0.4f!! v_0=%0.1f t=%d x_0=%0.2f \\nu=%0.3f',mm,v_0
        break;
    end;
end;
```

5.1 Take x(0) = 0.5 and v(0) = 0. Simulate the uncontrolled dynamics and vary the parameters such that 1) the problem is too easy and all trajectories reach the top of the hill and 2) the problem is not too hard that no trajectories reach the top of the hill

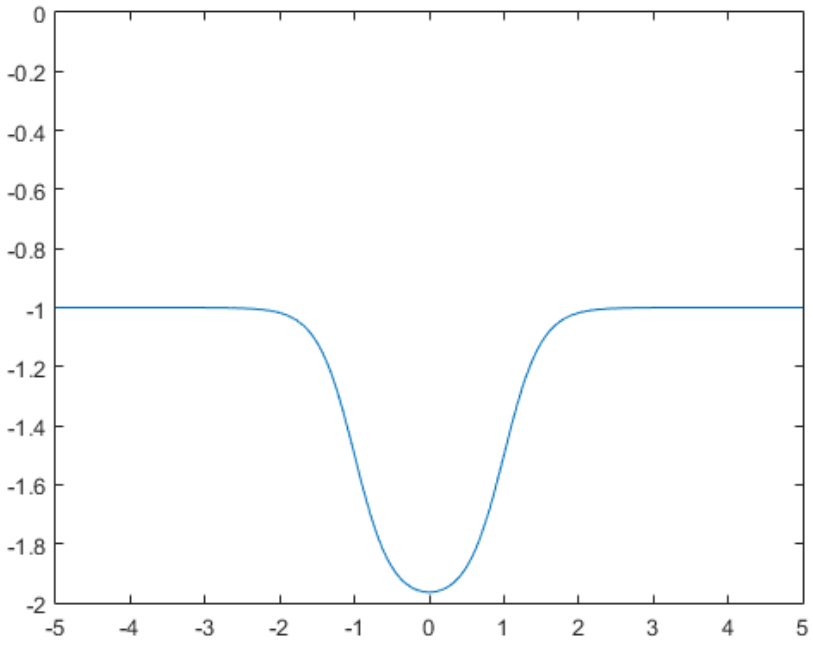

Figure 2: The shape of the valley.

5.1.1 1

To simulate the problem being easy we had to change gravitational acceleration $g = 0.1$ and we changed the noise to $\nu = 0.1$ which can most times cause the car to get out. The trajectory of the car can be seen in red. Our target time was T=10 and the time the car got out can be seen in the figure description. The time discretization we will be using is $dt = 0.001$

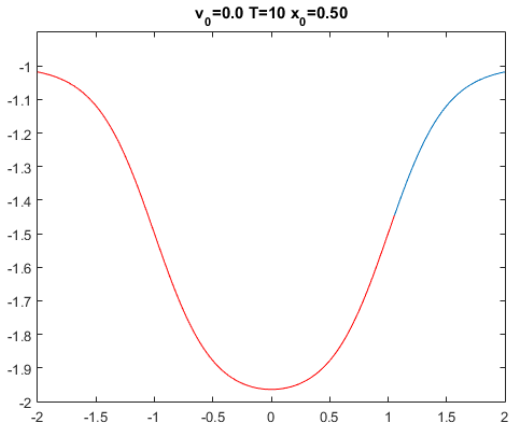

Figure 3: Made it out at x=-2.0095!! $v_0 = 0.0$
t=8.07 $x_0 = 0.50$ $\nu = 0.100$

5.1.2 2

By changing the gravitational acceleration to $g = 5$ (Mercury?) we made the problem harder but not too hard as can be seen in Figure 4. No trajectories can reach the top of the hill but they come close to the middle.

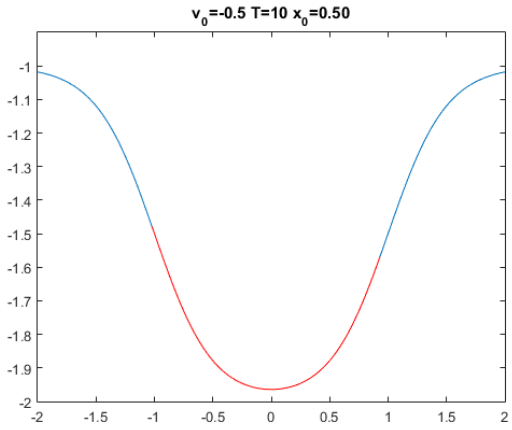

Figure 4: Cant get out.

5.2 With the parameter values found above, compute the optimal cost to go.

We will be treating $\phi(x(T))/\lambda$ as an energy function and we will be trying to sample from the distribution $exp(-\phi(x(T))/\lambda)$. We will be using a Markov chain of 200 samples as can be seen in the code. The target will happen at time T=10.

```
d_t=0.01
plot_=0;
for x_0=[0]
    for T=[1,10,50,100]
        for nu=[0.01,0.1,0.5]
```

```
            plot_=plot_+1;
            time_discretation=0:d_t:T-d_t;
            X_ALL=[];
            DU_ALL=[];
            x=x_0;
            u=0;
            for t=time_discretation %move through time and calculate the steps
                u_star=(tanh(x/(nu*(T-t)))-x)/(T-t);
                d_x=u_star*d_t+normrnd(0,sqrt(nu*d_t));
                DU_ALL=[DU_ALL u_star];
                x=x+ d_x;
                X_ALL=[X_ALL x];
            end;
            ax=subplot(4,3,plot_);
            plot(time_discretation,X_ALL,'k',time_discretation,DU_ALL,'r');
            hline=refline(ax,[0 1]);
            hline=refline(ax,[0 -1]);
            title(sprintf('x0=%0.1f T=%d \\nu=%0.2f',x_0,T,nu));
        end;
    end;
end;
```

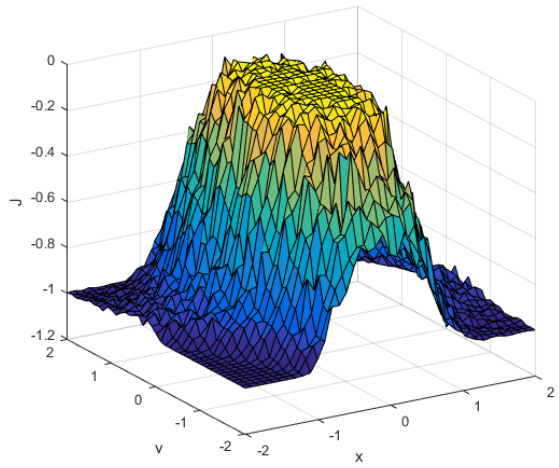

Figure 5:

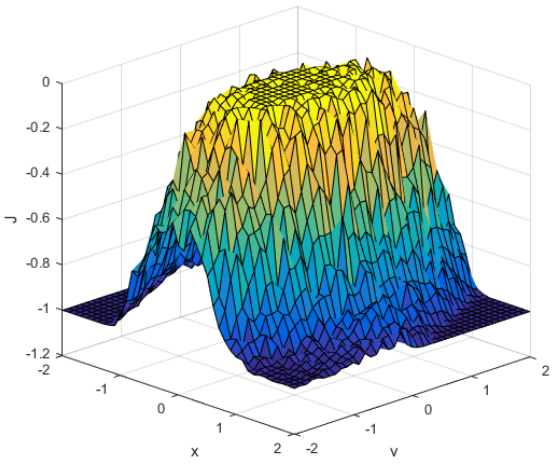

Figure 6:

As we can see the cost to go depends heavily on the starting x_0 and a good starting point is usualy closer to the sides +/-2 as expected where the J is minimal. The cost-to-go takes its biggest values near $x_0 = 0$ and $v_0 = 0$. J depends less heavily on the speed, this might have to do with our choice of g. We can see the dependence on speed at the small ripple up from v[0..2],x[-2] in figure 5. We can see the same ripple on the other side in figure 6. This is expected since it would be nice to have initial speed with different sign than the initial x. And if you dont have that even with x_0 on the sides you could fall back in and be trapped.
A more clear overview of the speed effect is shown below in figure 7. We see that even for x=0 high values of speed have the effect of lowering the cost-to-go.

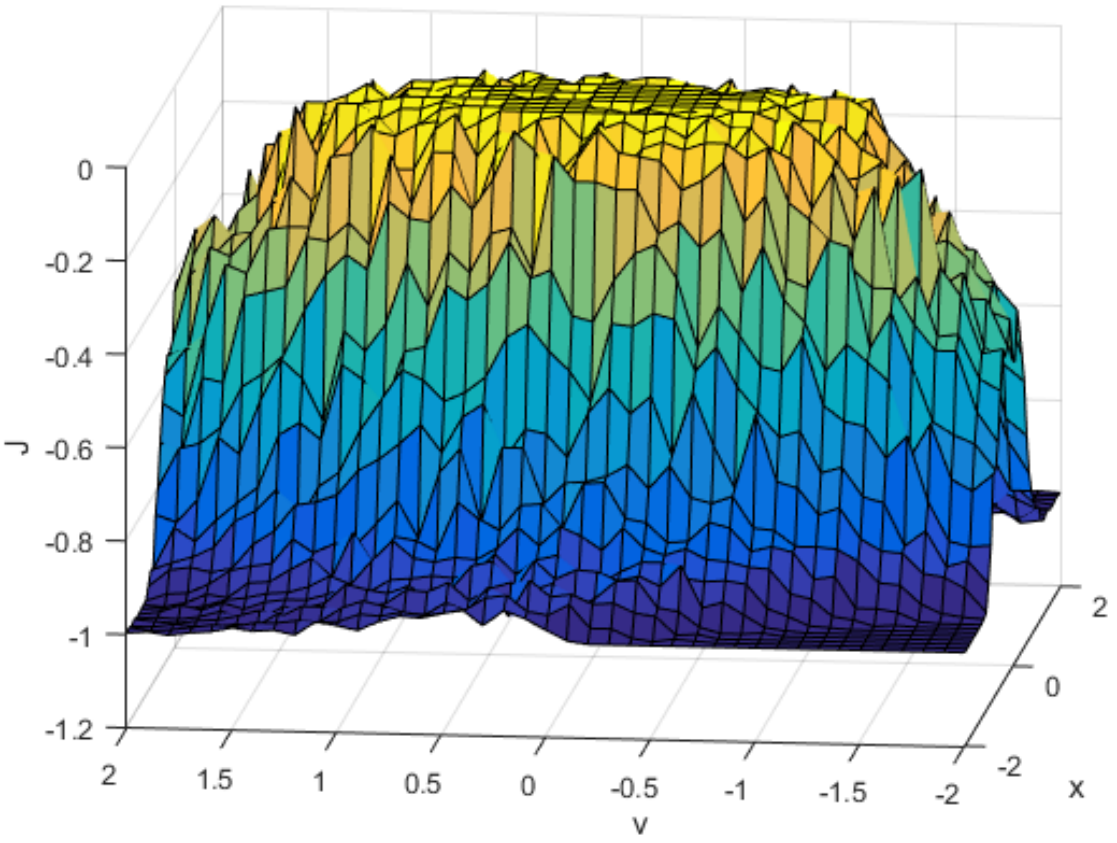

Figure 7: