Gradient Descent Methods on Optimal Control Problems

Enrique Zuazua

FAU - AvH

Based on the presentation by Dongnam Ko at the EECI Summer School, SCU, Chengdu, China, July 2019

April 20, 2020

Table of Contents

Example : a linear control problem

Recall the open-loop linear control problems.

- Equation of motion : The state $x(t)$ and control $u(t)$ satisfy

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \quad t \in [0, T], \\ x(0) = x^0 \in \mathbb{R}^n, \end{cases}$$

where $u : [0, T] \to U \subset \mathbb{R}^m$.

- The control objective : $x(T) = x^1 \in \mathbb{R}^n$.
- Kalman rank condition : $\text{rank}[B, AB, \cdots, A^{n-1}B] = n$.

Example : a linear control problem

Then, we used adjoint system of the costate $\phi(\cdot)$ and built an optimization process to find an open-loop control u.

- Adjoint system :

$$\begin{cases} -\dot{\phi}(t) = A^*\phi(t), \quad t \in [0, T], \\ \phi(T) = \phi^T \in \mathbb{R}^n, \end{cases}$$

A control from optimization

Then, the open-loop control is given by $u^*(t) = B^*\phi(t)$,

$$J(\phi_T) = \frac{1}{2}\int_0^T |B^*\phi|^2 dt + \langle x^0, \phi(0)\rangle.$$

Example : a linear control problem

One of the common ways to find the minimizer ϕ^T is the Gradient Descent method.

- Optimal problem : Find $\phi^T \in \mathbb{R}^d$ which minimizes

$$J(\phi^T) = \frac{1}{2}\int_0^T |B^*\phi(t)|^2 dt + \langle x^0, \phi(0)\rangle.$$

- From an initial guess on ϕ_0^T, use an iterative process for small $\alpha > 0$:

$$\phi_{k+1}^T := \phi_k^T - \alpha \nabla_{\phi_k^T} J(\phi_k^T), \quad k = 0, 1, \cdots .$$

Example : a linear control problem

How can we calculate the gradient, $\nabla_{\phi_k^T} J(\phi_k^T)$?

- The costate $\phi(t)$ is the solution of the adjoint system from the final datum ϕ^T,

$$\phi(t) = e^{-A^*(T-t)}\phi^T, \quad \phi(0) = e^{-A^* T}\phi^T.$$

- Then, the cost function becomes

$$J(\phi^T) = \frac{1}{2}\int_0^T |B^* e^{-A^*(T-t)}\phi^T|^2 dt + \langle x^0, e^{-A^* T}\phi^T\rangle.$$

- Now we can differentiate J in terms of ϕ^T.

Table of Contents

Formulation of optimal control problems

- Equation of motion :

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)), \quad t \in [0, T], \\ x(0) = x_0 \in \mathbb{R}^n, \end{cases} \tag{1}$$

where $u : [0, T] \to U \subset \mathbb{R}^m$.

- The control cost : $J(x(\cdot), u(\cdot))$.

- Problem : Find a open-loop control $u(t)$, which minimizes the control cost for a controlled system.

$$u^*(t) = \text{argmin}\{J(x(\cdot), u(\cdot)) \mid u : [0, T] \to U\},$$

subject to the equation (1).

Gradient descent method

- Since the control $u : [0, T] \to U$ determines the state $x(t, u(\cdot))$, we need to calculate the derivative

$$\frac{\partial}{\partial u(\cdot)} J(x(\cdot, u(\cdot)), u(\cdot)).$$

- Discretization of the time : For $0 = t_0 < t_1 < \cdots < t_N = 1$, the states and control can be represented by $x_n = x(t_n)$ and $u_n = u(t_n)$, for example, we may use the forward Euler method:

$$x_{k+1} = x_k + (t_{k+1} - t_k) f(x_k, u_k), \quad k = 0, 1, \cdots .$$

- Then, the problem becomes

$$\min_{(u_{\mathbf{0}}, \cdots, u_N)} \bar{J}(u_0, \cdots, u_N)$$
$$= \min_{(u_{\mathbf{0}}, \cdots, u_N)} J(x_0(u_0, \cdots, u_N), \cdots, x_N(u_0, \cdots, u_N), u_0, \cdots, u_N).$$

Gradient descent method

The calculation of the gradient (total derivative) on

$$\bar{J}(u_0, \cdots, u_N) = J(x_0(u_0, \cdots, u_N), \cdots, x_N(u_0, \cdots, u_N), u_0, \cdots, u_N)$$

is a tough problem. There are two common options to operate it.

1. Minimizing a function with constraints :
   Minimize the cost function over both the state and the control,

   $$\min_{x_1, \cdots, x_N, u_0, \cdots, u_N} J(x_0, \cdots, x_N, u_0, \cdots, u_N),$$

   with the equation of motion as constraints,

   $$x_{k+1} - x_k - (t_{k+1} - t_k) f(x_k, u_k) = 0, \quad k = 0, 1, \cdots.$$

2. Adjoint approach : We may calculate the gradient of the cost function using the adjoint system.

Adjoint approach

- Equation of motion :

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)), & t \in [0, T], \\ x(0) = x_0 \in \mathbb{R}^n, \end{cases} \tag{2}$$

where $u : [0, T] \to U \subset \mathbb{R}^m$.

- The control cost : $J = \Psi(x(T)) + \int_0^T L(x(t), u(t))dt$.

- Problem : Find a open-loop control $u(t)$, which minimizes the control cost for a controlled system.

$$u^*(t) = \text{argmin}\{J(x(\cdot), u(\cdot)) \mid u : [0, T] \to U\},$$

subject to the equation (7).

Adjoint approach

- The control cost can be understood as an Euler-Lagrange problem,

$$\text{Minimize} \quad J(x(\cdot), u(\cdot)) = \Psi(x(T)) + \int_0^T L(x(t), u(t))dt,$$

subject to the constraints

$$x(0) - x^0 = 0,$$
$$\dot{x}(t) - f(x(t), u(t)) = 0, \quad t \in [0, T].$$

- We adopt the Lagrange multiplier λ (the 'momentum') and the Lagrangian $\mathcal{L}$,

$$\mathcal{L}(x, u, \lambda) = \int_0^T (L(x, u) - \lambda \cdot (\dot{x} - f(x, u)))\, dt + \Psi(x(T))$$
$$= \int_0^T (H(x, u, \lambda) - \lambda \cdot \dot{x})\, dt + \Psi(x(T)).$$

Adjoint approach

- Then, from the optimality of the state, we may consider the derivative along with δx,

$$\begin{aligned}\delta \mathcal{L} &= \int_0^T (H_x \cdot \delta x - \lambda \cdot \delta \dot{x})\, dt + \Psi_x \cdot \delta x(T) \\ &= \int_0^T \left((H_x + \lambda) \cdot \delta x - \frac{d}{dt}(\lambda \cdot \delta x) \right) dt + \Psi_x \cdot \delta x(T) \\ &= \int_0^T (H_x + \lambda) \cdot \delta x dt + (\Psi_x - \lambda(T)) \cdot \delta x(T).\end{aligned}$$

- This implies the adjoint system with respect to the Hamiltonian H:

$$\begin{cases} -\dot{\lambda} = H_x(x, u, \lambda) = L_x(x, u) + f_x(x, u) \cdot \lambda, \\ \lambda(T) = \Psi_x(x(T)), \end{cases}$$

where the solution of the system for (x, u, λ) will be the optimal trajectories to minimize $\mathcal{L}$, i.e., J with constraints.

Pontryagin Maximal Principle

Pontryagin Maximal Principle (PMP)

Define the Hamiltonian of the Lagrangian L,

$$H(x, u, \lambda) := L(x, u) + f(x, u) \cdot \lambda.$$

Then, if $\bar{x}(t)$, $\bar{u}(t)$ are the optimal state and control trajectories, then there exists a costate $\bar{\lambda}(t)$ satisfying

$$\begin{cases} -\dot{\bar{\lambda}} = L_x(\bar{x}, \bar{u}) + f_x(\bar{x}, \bar{u}) \cdot \bar{\lambda}, \\ \bar{\lambda}(T) = \Psi_x(\bar{x}(T)), \end{cases}$$

where the optimal control $u(t)$ satisfies

$$\bar{u} = \text{argmin}_u H(\bar{x}, u, \bar{\lambda}).$$

Gradient descent using the adjoint system

- Note that PMP requires us to find the optimal state, control and costate simultaneously. (A system of ordinary differential equations with boundary values.)

- Instead, we can follow an iterative method using the gradient descent method.

1. From an initial guess u_0 on $u(t)$, we may define x_0 and λ_0,

$$\dot{x}_0 = f(x_0, u_0), \quad t \in [0, T], \quad x_0(0) = x^0 \in \mathbb{R}^n,$$

$$\dot{\lambda}_0 = L_x(x_0, u_0) + f_x(x_0, u_0) \cdot \lambda_0, \quad t \in [0, T], \quad \lambda_0(T) = \Psi_x(x_0(T)).$$

2. From x_k, u_k and λ_k, we have the gradient of the Hamiltonian,

$$H_u(x_k, u_k, \lambda_k) = L_u(x_k, u_k) + f_u(x_k, u_k) \cdot \lambda_k,$$

H_u is the same as the gradient of J

$$\frac{d}{du(\cdot)} J(x(\cdot, u(\cdot)), u(\cdot))\Big|_{u(\cdot)=u_k} = H_u(x_k, u_k, \lambda_k).$$

Example : a harmonic oscillator

- Equation of motion :

$$\begin{cases} \ddot{x}(t) + x(t) = u(t), \quad t \in [0, T], \quad T = \pi, \\ x(0) = 1, \quad \dot{x}(0) = 0. \end{cases} \tag{3}$$

where $u : [0, T] \to \mathbb{R}^1$.

- Let $y(t) = (x(t), \dot{x}(t))$.
- The control cost : $J = \frac{1}{2}(|x(T)|^2 + |\dot{x}(T)|^2) + \frac{1}{2}\int_0^T |u(t)|^2 dt$, where

$$\Psi(y(T)) = \frac{1}{2}|y(T)|^2 \quad \text{and} \quad L(y(t), u(t)) = \frac{1}{2}|u(t)|^2.$$

- Problem : Find the gradient of J at $u_0 = 0$ with respect to u.

Example : a harmonic oscillator

- Discretization of the time : For $0 = t_0 < t_1 < \cdots < t_N = 1$, the states and control can be represented by $x_n = x(t_n)$, $\dot{x}_n = \dot{x}(t_n)$ and $u_n = u(t_n)$,

$$x = (x_0, x_1, \ldots, x_N), \;\; \dot{x} = (\dot{x}_0, \dot{x}_1, \ldots, \dot{x}_N), \;\; u = (u_0, u_1, \ldots, u_N).$$

- Then, the problem becomes

$$\min_{(x,\dot{x},u)} J(x, \dot{x}, u),$$

subject to the equation of motion.

- For example, we may use the forward Euler method:

$$\begin{aligned} x_{k+1} &= x_k + (t_{k+1} - t_k)\dot{x}_k, \\ \dot{x}_{k+1} &= \dot{x}_k + (t_{k+1} - t_k)(u_k - x_k), \\ x_0 &= 1, \quad \dot{x}_0 = 0. \end{aligned}$$

Example : a harmonic oscillator

We may use the adjoint system to calculate gradient.

- The running cost L and the final cost Ψ are

$$L(y(t), u(t)) = \frac{1}{2}u(t)^2 \quad \text{and} \quad \Psi(y(1)) = \frac{1}{2}(x(1)^2 + \dot{x}(1)^2).$$

- Then, the adjoint system is

$$-\dot{\lambda} = L_y + f_y \cdot \lambda = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \lambda,$$
$$\lambda(1) = \Psi_y(y(1)) = y(1).$$

- Now we may calculate the gradient from the state, control and costate, y, u and λ:

$$H_u(x, u, \lambda) = L_u + f_u \cdot \lambda = u(t) + (0, 1) \cdot \lambda(t).$$

Example : a harmonic oscillator

- Let $u_0 = 0$. The corresponding $y_0 = (x_0, \dot{x}_0)$ satisfies

$$\ddot{x}_0(t) + x_0(t) = u_0(t) = 0.$$

Then, from $y_0(0) = (1, 0)$, we have $y_0(t) = (\cos t, -\sin t)$.

- From the adjoint equation

$$-\dot{\lambda} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \lambda, \quad \lambda(\pi) = y(\pi),$$

we have $\lambda_0(\pi) = y_0(\pi) = (-1, 0)$. Then,

$$\lambda_0(t) = (\cos t, \sin t).$$

- Finally, the gradient becomes

$$\begin{aligned} H_u(x_0, u_0, \lambda_0) &= L_u(x_0, u_0) + f_u(x_0, u_0) \cdot \lambda_0 \\ &= 0 + (0, 1) \cdot (\cos t, \sin t) \\ &= \sin t. \end{aligned}$$

Example : a harmonic oscillator

We may compare $H_u = \sin t$ with the gradient of the cost J.

- Note that $J = \frac{1}{2}(|x(T)|^2 + |\dot{x}(T)|^2) + \frac{1}{2}\int_0^T |u(t)|^2 dt$. Then,

$$\frac{dJ}{du}\bigg|_{u=u_0} = u_0(t) + y_0(\pi) \cdot \frac{dy(\pi)}{du}\bigg|_{u=u_0}.$$

- The derivative of the final state with respect to the control function:

$$\ddot{\delta x} + \delta x = \delta u \quad \text{and} \quad \delta y(0) = (0,0), \quad \text{find} \quad \delta y(\pi).$$

- For the Dirac delta function $\delta u = \delta_0(t)$, we have $\delta x = (\sin t, \cos t)$. In the same way, for $\delta u = \delta_{t_0}(t)$,

$$\delta x(\pi) = (\sin(\pi - t_0), \cos(\pi - t_0)) = (\sin t_0, -\cos t_0).$$

Example : a harmonic oscillator

- We conclude that

$$\left\langle \frac{dJ}{du}\Big|_{u=u_0}, \delta_{t_0}(t) \right\rangle = u_0 + y_0(\pi) \cdot \left\langle \frac{dy(\pi)}{du}\Big|_{u=u_0}, \delta_{t_0}(t) \right\rangle$$
$$= 0 + (-1, 0) \cdot (\sin t_0, -\cos t_0) = -\sin t_0.$$

- Hence, the total derivative on J is the same as the partial derivative of H in L^∞.

- Now, the next iteration starts from

$$u_1 = u_0 - \alpha H_u(x_0, u_0, \lambda_0),$$

with proper $\alpha > 0$.

Table of Contents

Computational Platform: DyCon Toolbox

DyCon Toolbox is a software platform developed in MATLAB which implements a set of tools to solve mathematical problems of Optimal Control. It's goal is to provide a software architecture that allows modular algorithms to be integrated, in addition to providing visualization tools.

DyCon Toolbox

https://deustotech.github.io/dycon-platform-documentation/

Computational Platform: DyCon Toolbox

DyCon Toolbox is developed around the minimum principle of Pontryagin. Thanks to the symbolic MATLAB engine, problems can be defined in a general way. For example, the following problem:

$$J = \|Y(T) - Y_T\|^2 + \frac{1}{2}\int \|U(t)\|^2 dt \ \text{ subject to } \ \dot{Y} = AY + BU,$$

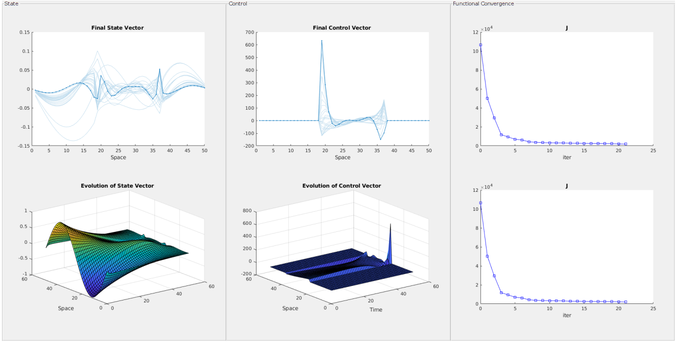
can be stated in DyCon Toolbox in few lines of code

```
>> Y = sym('y',[2 1]); U = sym('u',[2 1]);
>> odeDyn = ode(@(t,Y,U,P) A*Y+B*U,Y,U,sym.empty);
>> numPsi = @(T,Y)    ([1,1] - Y).'*([1,1] - Y);
>> numL   = @(t,Y,U)  0.5*(U.'*U);
>> iCP1 = Pontryagin(odeDyn,numPsi,numL);
```

Computational Platform: DyCon Toolbox

These can be resolved througout different optimization methods provided by DyCon Toolbox and other external libraries. For example:

```
>> U0 = zeros(iCP1.Dynamics.Nt,iCP1.Dynamics.
    ControlDimension);
>> GradientMethod(iCP1,U0);
```



Computational Platform: DyCon Toolbox

DyCon Toolbox also provides a web platform, in which plenty of documentation on how to get started with the toolbox is available. It also provides further tutorials, practical examples, as well as a detailed installation guide.

DyCon Toolbox

3. The cost of control will be related to the collective dynamics we want, such as the variance of frequencies or phases.

HOME | DOCUMENTATION | FIRST STEPS | EXAMPLES | DOWNLOADS | CHAIR OF COMPUTATIONAL MATHEMATICS

Numerical simulation

Here, we consider a simple problem: we control the all-to-all network system to get gathered phases at final time T. We first need to define the system of ODEs in terms of symbolic variables.

```
clc
```

```
m = 5;  %% [m]: number of oscillators.

syms t;
symTh = sym('y', [m,1]);  %% [y]: phases of oscillators, $\theta_i$.
symOm = sym('om', [m,1]);  %% [om]: natural frequencies of osc., $\omega_i$.
symK = sym('K',[m,m]); %% [K]: the coupling network matrix, $\kappa$.
symU = sym('u',[1,1]); %% [u]: the control functions along time, $u(t)$.

syms Vsys;   %% [Vsys]: the vector fields of ODEs.
symThth = repmat(symTh,[1 m]);
Vsys = symOm + (symU./m)*sum(symK.*sin(symThth.' - symThth),2);   %% Kuramoto interaction terms.
```

The parameter ω_i and κ should be specified for the calculations. Practically, $K > |\max\Omega - \min\Omega|$ leads to the synchronization of frequencies. We normalize the coupling strength to 1, and give random values for the natural frequencies from the normal distribution $N(0, 0.1)$. We also choose initial data from $N(0, pi/4)$.

```
%% Om_init = normrnd(0,0.1,m,1);
%% Om_init = Om_init - mean(Om_init); %% Mean zero frequencies
%% Th_init = normrnd(0,pi()/4,m,1);
```

Example 1 : a harmonic oscillator

- Equation of motion :

$$\begin{cases} \ddot{x}(t) + x(t) = u(t), \quad t \in [0, T], \quad T = \pi, \\ x(0) = 1, \quad \dot{x}(0) = 0. \end{cases} \tag{4}$$

where $u : [0, T] \to \mathbb{R}^1$.

- Let $y(t) = (x(t), \dot{x}(t))$.
- The control cost : $J = \frac{100}{2}(|x(T)|^2 + |\dot{x}(T)|^2) + \frac{1}{2}\int_0^T |u(t)|^2 dt$, where

$$\Psi(y(T)) = \frac{1}{2}|y(T)|^2 \quad \text{and} \quad L(y(t), u(t)) = \frac{1}{2}|u(t)|^2.$$

- Problem : Find $u^*(t)$, which minimizes J under (5).

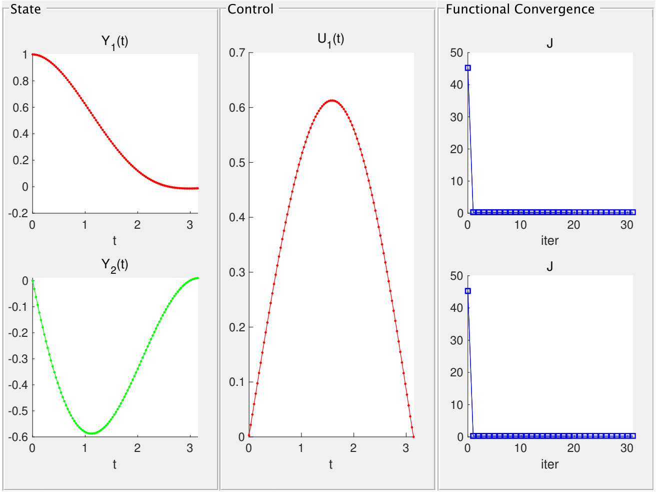
Example 1 : a harmonic oscillator

Then, the code can be built as follows,

```
>> A = [ 0 1 ; -1 0]; B = [0 ; 1];
>> odeDyn = ode('A',A,'B',B,'Nt',100,'FinalTime',pi);
>> odeDyn.InitialCondition = [1;0];
>> numPsi = @(T,Y) 50*Y.'*Y;
>> numL = @(t,Y,U) 0.5*U.'*U;
>> iCP = Pontryagin(odeDyn,numPsi,numL);
>> U0 = zeros(iCP.Dynamics.Nt,iCP.Dynamics.ControlDimension);
>> GradientMethod(iCP,U0);
```

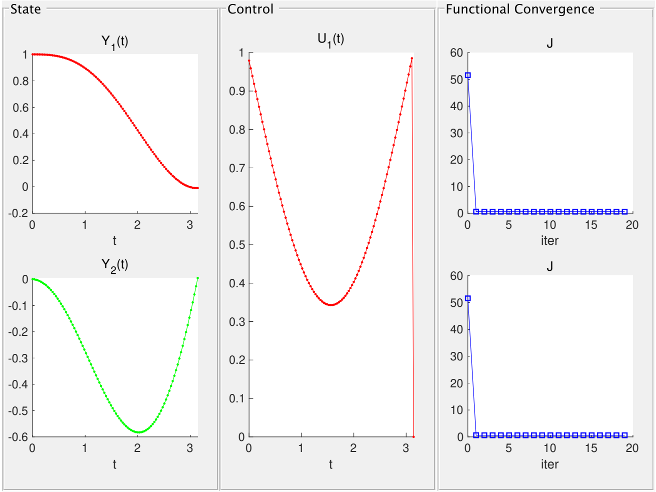
Example 1 : a harmonic oscillator

Initial guess of the control : constant zero



Example 1 : a harmonic oscillator

Initial guess of the control : constant one



Example 2 : a heat equation

- Equation of motion : For $x \in [-1, 1]$ and $t \in [0, 0.1]$,

$$\begin{cases} y'(t,x) - \Delta y(t,x) = u(t,x) 1_{[-1/2,1/2]}(x), \\ y(0,x) = \sin((\pi/2)x), \quad \dot{x}(t,-1) = x(t,1) = 0. \end{cases} \tag{5}$$

- Goal of the control : $y(0.1, x) \simeq y^T = 0$.

- The control cost : $J = \frac{10^{12}}{2}(\|y(0.1) - y^T\|^2) + \int_0^T \|u(t)\| dt$.

- Problem : Find $u^*(t)$, which minimizes J.

Example 2 : a heat equation

```
N = 20;
xi = -1; xf = 1;
xline = linspace(xi,xf,N+2);
xline = xline(2:end-1);
dx = xline(2) - xline(1);
A = FDLaplacian(xline);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
a = -0.5; b = 0.5;
B = BInterior(xline,a,b,'min',false);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
FinalTime = 0.1;
dt = 0.001;
Y0 =sin(0.5*pi*xline');

dynamics = pde('A',A,'B',B,'InitialCondition',Y0,'
    FinalTime',FinalTime,'Nt',5);
dynamics.mesh= xline;
```

Example 2 : a heat equation

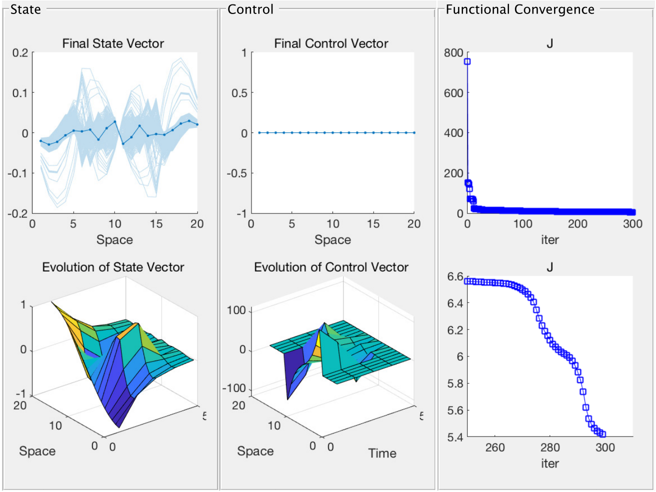
```
Y = dynamics.StateVector.Symbolic;
U = dynamics.Control.Symbolic;

YT = 0*cos(0.5*pi*xline');
epsilon = dx^4;
symPsi  = @(T,Y)    dx*(1/(2*epsilon))*(YT - Y).'*(YT - Y);
symL    = @(t,Y,U)  dx*sum(abs(U));

iCP1 = Pontryagin(dynamics,symPsi,symL);
tol = 1e-8;
U0 = zeros(iCP1.Dynamics.Nt,iCP1.Dynamics.ControlDimension);
[UOptDyCon,JOptDycon] = GradientMethod(iCP1,U0,'tol',tol,'Graphs',true,'DescentAlgorithm',@ConjugateDescent,'MaxIter',300,'display','all')
```

Example 2 : a heat equation

Iteration : 300



Example 2 : a heat equation

Iteration : 600

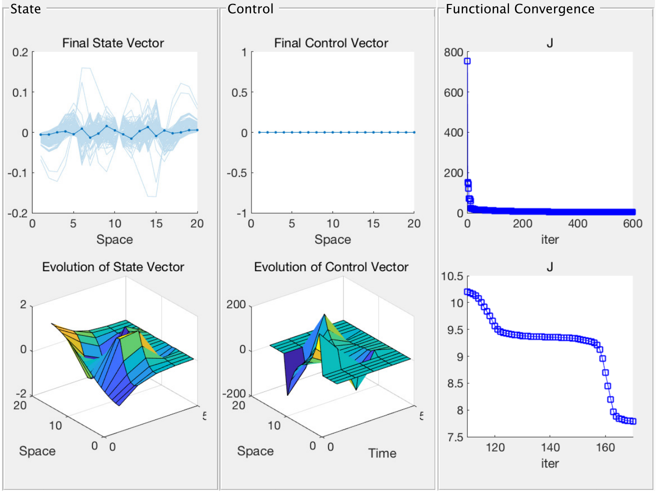


Table of Contents

The herding problem : Shepherd dogs and sheep

The number of individuals is small, yet the interaction dynamics and control strategies is complex

We consider the "guidance by repulsion" model based on the two-agents framework: the driver tries to drive the evader.

The drivers want to control the evaders:

1. Gathering of the evaders,
2. Driving the evaders into a desired area.

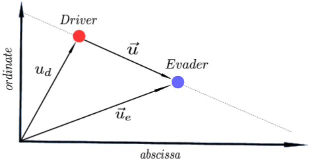


Figure: Picture of Border Collie [from Wikipedia] and the diagram of the model

Motivation: "Guidance by repulsion" model

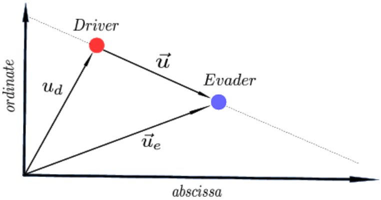
R. Escobedo, A. Ibañez and E.Zuazua, Optimal strategies for driving a mobile agent in a "guidance by repulsion" model, Communications in Nonlinear Science and Numerical Simulation, 39 (2016), 58-72.

[R. Escobedo, A. Ibañez, E. Zuazua, 2016] suggested a **guidance by repulsion** model based on the two-agents framework: *the driver*, which tries to drive the *evader*.

1. The driver follows the evader but cannot be arbitrarily close to it (because of chemical reactions, animal conflict, etc).
2. The evader moves away from the driver but doesn't try to escape beyond a not so large distance.
3. The driver is faster than the evader.
4. At a critical short distance, the driver can display a **circumvention maneuver** around the evader, forcing it to change the direction of its motion.
5. By adjusting the circumvention maneuver, the evader can be driven towards a desired target or along a given trajectory.

One sheep + one dog + Circumvention control

The control $k(t)$ is chosen in feedback form to align the gate, the sheep and the dog.



In short, the model for $\mathbf{u}_d, \mathbf{u}_e \in \mathbf{R}^2$ can be written with nonlinear interaction kernels $f_d(\cdot)$ and $f_e(\cdot)$:

$$\begin{cases} \dot{\mathbf{u}}_d = \mathbf{v}_d, \quad \dot{\mathbf{u}}_e = \mathbf{v}_e \\ m_d \dot{\mathbf{v}}_d = -f_d(|\mathbf{u}_d - \mathbf{u}_e|)(\mathbf{u}_d - \mathbf{u}_e) - \nu_d \mathbf{v}_d + \kappa(t)(\mathbf{u}_d - \mathbf{u}_e)^{\perp} \\ m_e \dot{\mathbf{v}}_e = -f_e(|\mathbf{u}_e - \mathbf{u}_d|)(\mathbf{u}_d - \mathbf{u}_e) - \nu_e \mathbf{v}_e \\ \mathbf{u}_d(0) = \mathbf{u}_d^0, \ \mathbf{u}_e(0) = \mathbf{u}_e^0, \ \mathbf{v}_d(0) = 0, \ \mathbf{v}_e(0) = 0 \end{cases} \tag{6}$$

Studies on the herding problem

In this setting, they considered bang-bang type controls with open-loop and feed-back strategies.

Similar consideration have been addressed with repulsive interactions in control theory:

- Defender-intruder strategy : [Wang, Biegler, 2006],
- Hunting strategy model :
  [Muro, Escobedo, Spector, Coppinger, 2011 and 2014],
- Dog-sheep gathering problem :
  Well-posedness of optimal control problems [Burger, Pinnau, Roth, Totzeck, Tse, 2016]
  and its simulations [Pinnau, Totzeck, 2018].

Guidance-by-repulsion model with many individuals

Let $\mathbf{u}_{dj}, \mathbf{u}_{ei} \in \mathbb{R}^2$ are positions of drivers and evaders for $i = 1, \cdots, N$ and $j = 1, \cdots, M$.

When there are many evaders, we need to suggest a representative position of evaders which the drivers follow. We set the barycenter of evaders,

$$\mathbf{u}_{ec} := \frac{1}{N} \sum_{k=1}^{N} \mathbf{u}_{ek},$$

then the dynamics can be described by

$$\begin{cases} \ddot{\mathbf{u}}_{dj} = -f_d(|\mathbf{u}_{dj} - \mathbf{u}_{ec}|)(\mathbf{u}_{dj} - \mathbf{u}_{ec}) - \nu \dot{\mathbf{u}}_{dj} + \kappa_j(t)(\mathbf{u}_{dj} - \mathbf{u}_{ec})^{\perp}, \\ \ddot{\mathbf{u}}_{ei} = -\dfrac{1}{M} \displaystyle\sum_{j=1}^{M} f_e(|\mathbf{u}_{dj} - \mathbf{u}_{ei}|)(\mathbf{u}_{dj} - \mathbf{u}_{ei}) \\ \qquad\quad -\dfrac{1}{N} \displaystyle\sum_{k=1}^{N} f_g(|\mathbf{u}_{ek} - \mathbf{u}_{ei}|)(\mathbf{u}_{ek} - \mathbf{u}_{ei}) - \nu \dot{\mathbf{u}}_{ei}, \\ \mathbf{u}_{dj}(0) = \mathbf{u}_{dj}^0, \ \mathbf{u}_{ei}(0) = \mathbf{u}_{ei}^0, \ \dot{\mathbf{u}}_{dj}(0) = \mathbf{v}_{dj}^0, \ \dot{\mathbf{u}}_{ei}(0) = \mathbf{v}_{ei}^0. \end{cases}$$

First order reduced model with one driver and one evader

From now on, we consider one driver and one evader model for analytic results.

For simplicity, we first observe the dynamics of its reduced limit, $m_e, m_d \to 0$. This singular limit removes the effect of inertia, hence, we get the long-time behavior monotonically.

$$\begin{cases} \dot{\mathbf{u}}_d = \mathbf{v}_d, \quad \dot{\mathbf{u}}_e = \mathbf{v}_e \\ \nu_d \dot{\mathbf{u}}_d = -f_d(|\mathbf{u}_d - \mathbf{u}_e|)(\mathbf{u}_d - \mathbf{u}_e) + \kappa(t)(\mathbf{u}_d - \mathbf{u}_e)^{\perp} \\ \nu_e \dot{\mathbf{u}}_e = -f_e(|\mathbf{u}_e - \mathbf{u}_d|)(\mathbf{u}_d - \mathbf{u}_e) \\ \mathbf{u}_d(0) = \mathbf{u}_d^0, \ \mathbf{u}_e(0) = \mathbf{u}_e^0, \end{cases}$$

where the relative position $\mathbf{u} := \mathbf{u}_d - \mathbf{u}_e$ satisfies a closed equation,

$$\dot{\mathbf{u}} = -f(|\mathbf{u}|)\mathbf{u} + \kappa(t)\mathbf{u}^{\perp}.$$

From this relation, we can separately treat central velocity $-f(|\mathbf{u}|)\mathbf{u}$ and its perpendicular velocity $\kappa(t)\mathbf{u}^{\perp}$.

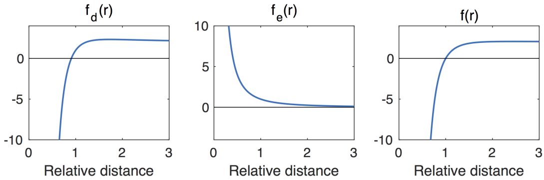
Interaction functions

Since we want the relative position $\mathbf{u}$ to satisfy the regulation between the driver and evader, we assume $f(r) = f_d(r) - f_e(r)$ satisfy

$$f(r) = \begin{cases} \geq 0 & \text{for} \quad r \geq r_c, \\ < 0 & \text{for} \quad 0 < r < r_c \end{cases} \quad \text{with} \quad f'(r_c) > 0,$$

which implies that $|\mathbf{u}|$ tends to r_c in the absence of control $\kappa(t)$.
As an example, we suggest

$$f_d(r) = \frac{2}{r^2} - \frac{3}{r^4} + 2 \quad \text{and} \quad f_e(r) = \frac{1}{r^2},$$



Potential function as a Lyapunov function

For the potential function

$$P(r) := \int_{r_c}^{r} sf(s)ds,$$

we may describe its gradient property:

$$\dot{\mathbf{u}} = -\nabla P(|\mathbf{u}|) + \kappa(t)\mathbf{u}^{\perp},$$

The potential function plays the role of Lyapunov function.

$$\begin{aligned}\dot{P}(|\mathbf{u}|) &= \frac{dP}{d|\mathbf{u}|} \cdot \frac{d|\mathbf{u}|}{dt} = |\mathbf{u}| f(|\mathbf{u}|) \frac{\langle \mathbf{u}, \dot{\mathbf{u}} \rangle}{|\mathbf{u}|} \\ &= f(|\mathbf{u}|)\langle \mathbf{u}, -f(|\mathbf{u}|)\mathbf{u} + \kappa(t)\mathbf{u}^{\perp} \rangle = -f(|\mathbf{u}|)^2 |\mathbf{u}|^2 \leq 0.\end{aligned}$$

Therefore, if we assume proper conditions on $f(r)$,

$$\int_0^{r_c} rf(r) = -\infty \quad \text{and} \quad \gamma_m := \liminf_{r\to\infty} f(r) > 0,$$

so that P is smooth, coercive, and blow-up at $r = 0$. Then, from the time derivative,

$$\dot{P}(|\mathbf{u}|) = -f(|\mathbf{u}|)^2|\mathbf{u}|^2 \leq 0.$$

we obtain dynamical properties.

Relative distance of the reduced model

- The relative distance $|\mathbf{u}|$ cannot be 0 from nonzero initial data, and uniformly bounded along time.
- $\mathbf{u}$ tends to the steady solution $\bar{\mathbf{u}}(t)$ which satisfies $f(|\bar{\mathbf{u}}|)|\bar{\mathbf{u}}| = 0$, that is,

$$|\mathbf{u}| \to r_c \quad \text{if} \quad |\mathbf{u}_0| \neq 0.$$

Note that the convergence is exponential since $f'(r_c) \neq 0$ so that $P(r)$ and $\dot{P}(r)$ are both quadratic on $f(r)$ locally.

Steady states and controllability

Finally, we may classify the steady states of $\mathbf{u}_d$ and $\mathbf{u}_e$.

- If $\kappa(t) \equiv 0$, then the dynamics is in a one-dimensional line including $\mathbf{u}_d^0$ and $\mathbf{u}_e^0$. Eventually, two agents tend to uniform linear motions.
- If $\kappa(t) \equiv 1$, then they converge to circular motions, where the relative distance is r_c and angular velocities are 1.

From these two states, we can control $\mathbf{u}_e$ into a desired position:

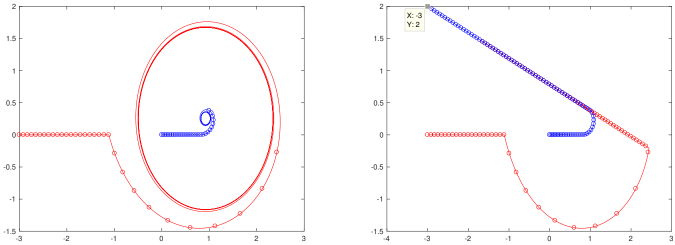


Figure: Rotational states (left) and off-bang-off control using it (right)

The Guidance-by-repulsion model

Next, we go back to the second order Guidance-by-repulsion model.

$$\ddot{\mathbf{u}} + f(|\mathbf{u}|)\mathbf{u} + \nu\dot{\mathbf{u}} = \kappa(t)\mathbf{u}^{\perp}.$$

For the interaction coefficient $f(r)$, we assume the same condition: for

$$P(r) := \int_{r_c}^{r} sf(s)ds \geq 0,$$

$P(0) = \infty$ and P grows quadratically $\left(\sim \frac{\gamma_m}{2}|\mathbf{u}|^2\right)$ as $r \to \infty$.

- The equation now follows the motion of damped oscillator under a central potential $P(|\mathbf{u}|)$ with an additional control term.
- The negativity/positivity of f makes the relative distance $\mathbf{u} \sim r_c$.

Two main regimes arise: Pursuit $\kappa(t) = 0$ / Circumvention $\kappa(t) \neq 0$.

Steady states

For each mode, we have the following steady states which characterize the dynamics:

- Pursuit mode: $\kappa(t) \equiv 0$:

$$\mathbf{u}(t) = \mathbf{u}_* \in \mathbb{R}^2 \quad \text{and} \quad \mathbf{v}(t) = (0,0) \quad \text{with} \quad |\mathbf{u}_*| = r_c,$$

where the driver and evader behave uniform linear motions,

$$\mathbf{u}_\ell(t) = -\frac{f_d(\mathbf{u}_*)\mathbf{u}_*}{\nu} t + \mathbf{u}_\ell(0), \quad \ell = d, e.$$

- Circumvention mode, $\kappa(t) \equiv \kappa$:

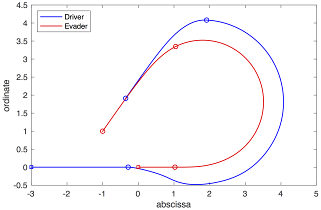
$$\mathbf{u}(t) = r_p \left(\cos\left(\frac{\kappa}{\nu} t\right), \sin\left(\frac{\kappa}{\nu} t\right) \right),$$

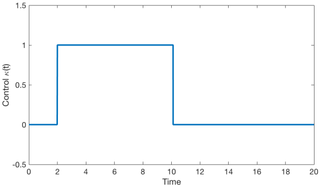
where the driver and evader have rotational motions on circles centered at the same point,

$$\mathbf{u}_\ell(t) = r_\ell \left(\cos\left(\frac{\kappa}{\nu} t + \phi_\ell\right), \sin\left(\frac{\kappa}{\nu} t + \phi_\ell\right) \right) + \mathbf{u}^*, \quad \mathbf{u}^* \in \mathbb{R}^2,\ \ell = d, e.$$

Off-Bang-Off control of the evader

Combining these two modes, we can construct an Off-Bang-Off control: choose the direction by rotations in the circumvention mode, and drive the evaders to the target in the pursuit mode.





Theorem [K.-Zuazua (preprint)]

Let $f(r)$ be as before. Then, for a given destination $\mathbf{u}_f \in \mathbb{R}^2$ and $\mathbf{u}_0 \neq (0,0)$, there exist t_1, t_2, t_f and κ such that the control function

$$\kappa(t) = \begin{cases} \kappa & \text{if} \quad t \in [t_1, t_2], \\ 0 & \text{if} \quad t \in [0, t_1) \cup (t_2, t_f], \end{cases} \qquad \text{satisfies} \quad \mathbf{u}_e(t_f) = \mathbf{u}_f.$$

Stability to the steady states

In order to analyze the off-bang-off control, we need to show the asymptotic stability to the steady states on each constant $\kappa(t)$.

The equation of the relative position $\mathbf{u}$ with constant control $\kappa(t) \equiv \kappa$,

$$\ddot{\mathbf{u}} + f(|\mathbf{u}|)\mathbf{u} + \nu\dot{\mathbf{u}} = \kappa\mathbf{u}^{\perp}, \quad \mathbf{u} \in \mathbf{R}^2,$$

which is the damped potential oscillator with an external source term.

However, the standard energy,

$$E(t) := \frac{1}{2}|\mathbf{v}|^2 + P(|\mathbf{u}|),$$

is no more non-increasing from the perpendicular term $\kappa(t)\mathbf{u}^{\perp}$.

$$\begin{aligned}\dot{E}(t) &= \mathbf{v} \cdot \dot{\mathbf{v}} + f(|\mathbf{u}|)\mathbf{u} \cdot \dot{\mathbf{u}} \\ &= \mathbf{v} \cdot (-f(|\mathbf{u}|)\mathbf{u} - \nu\mathbf{v} + \kappa(t)\mathbf{u}^{\perp}) + f(|\mathbf{u}|)\mathbf{u} \cdot \mathbf{v} \\ &= -\nu|\mathbf{v}|^2 + \kappa(t)\mathbf{u}^{\perp} \cdot \mathbf{v}.\end{aligned}$$

To fix it, we use hypocoercivity theory[1], and construct a perturbed energy using inner product terms:

$$L_{\pm}(t) = E(t) \pm \frac{\nu}{2}(\frac{\nu}{2}|\mathbf{u}|^2 + \mathbf{u}\cdot\mathbf{v}).$$

Then, its time derivative is

$$\dot{L}_{\pm}(t) \leq -\frac{\nu}{2}|\mathbf{v}|^2 + \frac{1}{2}(\nu f(|\mathbf{u}|) + \kappa(t))|\mathbf{u}|^2,$$

which is nonpositive if $|\mathbf{u}|$ is close to 0 or ∞.

On the other hand, if $\kappa(t)$ is constant, we may define κ dependent functions,

$$L_{\kappa}(t) = E(t) - \frac{\kappa}{\nu}\mathbf{u}^{\perp}\cdot\mathbf{v} \quad \text{and} \quad \dot{L}_{\kappa}(t) = -\nu\left|\mathbf{v} - \frac{\kappa}{\nu}\mathbf{u}^{\perp}\right|^2 \leq 0,$$

which is always nonpositive.

[1][C. Villani, 2009] and [K. Beauchard, E. Zuazua, 2011]

Therefore, we have the following dynamical properties.

Boundedness of relative distance

Suppose that the control $\kappa(t)$ is bounded: $\limsup_{t\to\infty} |\kappa(t)| < \nu\sqrt{\gamma_m}$.

Then, the relative position $\mathbf{u}(t)$ does not hit $(0,0)$ or blow-up in a finite time. Moreover, if $\kappa(t)$ is constant, then $\mathbf{u}(t)$ is uniformly bounded.

Global stability of steady states

The positions $\mathbf{u}_d(t)$ and $\mathbf{u}_e(t)$ converge to the steady states asymptotically if $\kappa(t) \equiv \kappa$ and $\kappa < \nu\sqrt{\gamma_m}$:

- If $\kappa = 0$, then $\mathbf{u}_d(t)$ and $\mathbf{u}_e(t)$ tend to linear motions.
- If $0 < |\kappa| < \nu\sqrt{\gamma_m}$, then $\mathbf{u}_d(t)$ and $\mathbf{u}_e(t)$ tend to rotational motions.

By combining these asymptotic steady states, we may prove the controllability of the evader's position to any desired point.

Since we can apply the Off-Bang-Off controls to any initial data, we may use it to pass multiple target points:

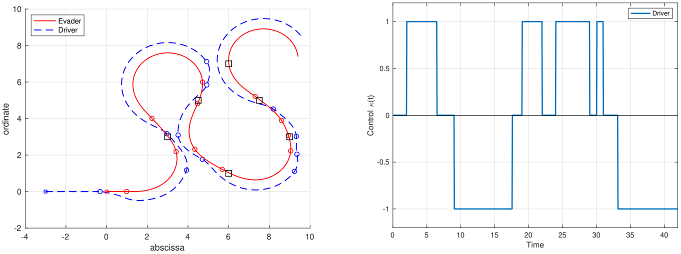


Figure: A trajectory of the evader which passes near points $(3,3)$, $(4.5,5)$, $(6,1)$, $(9,3)$, $(7.5,5)$ and $(6,7)$ denoted by black boxes.

This can be done by turning on and off $\kappa(t)$ using two control modes, where the dynamics converges to the corresponding steady state ('rotational motion' and 'linear motion') in a short time.

The effect of the number of evaders

If the evaders are gathered initially, the dynamics are similar to the one evader case, as we have one fat evader.

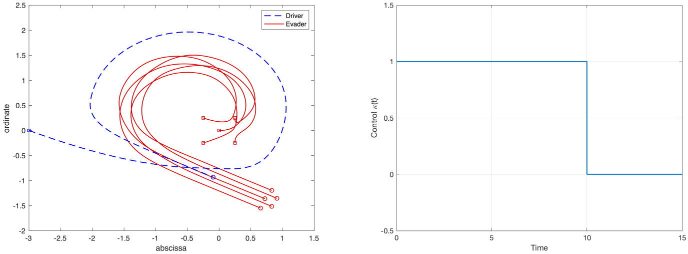


Figure: Trajectories of five evaders with a bang-off control $\kappa(t)$.

$$f_d(r) = \frac{2}{r^2} - \frac{3}{r^4} + 2, \quad f_e(r) = \frac{1}{r^2}, \quad \text{and} \quad f_g(r) = 10\left(\frac{(0.2)^2}{r^2} - \frac{(0.2)^4}{r^4}\right).$$

Table of Contents

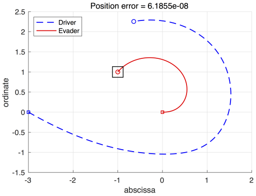
Optimal control strategies

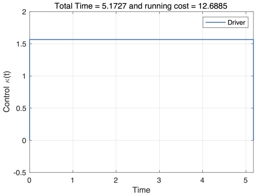
While Off-Bang-Off controls can drive the evaders properly, it is natural to find an optimal control strategy which minimizes a given cost.

For the cost function, we suggest two optimal control problems: On one hand, we want to minimize the running cost of controls:

$$J(\kappa(\cdot)) = \frac{1}{N}\sum_{i=1}^{N} |\mathbf{u}_{ei}(t_f) - \mathbf{u}_f|^2 + \frac{0.001}{M}\sum_{k=1}^{M}\int_0^{t_f} |\kappa_k(t)|^2 dt.$$

The simulations are done by gradient descent methods with flexible final time t_f, where the initial guess is given by constant control functions. For example, $\kappa(t) \equiv 1.5662$ and $t_f = 5.1727$ to make $\mathbf{u}(t_f) = (-1, 1)$:





Flexible final time

- Note : The control is to drive the evader to a specific position. The final state may not be a steady state!

- Since the final state is not a steady state, after a little time, it escapes the desired position. Then, this optimal control problem needs to have a flexible final time t_f.

- For example, let $\mathbf{u}_d(0) = (-1, 0)$, $\mathbf{u}_e(0) = (0, 0)$ and $\mathbf{u}_f = (1, 0)$ with initially zero velocities.
  Then, for a trivial control $\kappa(t) \equiv 0$, there is only one time t_f which satisfies $\mathbf{u}_e(t_f) = \mathbf{u}_f$.

- Therefore, the optimal control with a fixed time may not be a reasonable control.

Flexible final time

- The formulation of the Pontryagin maximum principle,

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)), \quad t \in [0, T], \\ x(0) = x_0 \in \mathbb{R}^n, \end{cases} \tag{7}$$

  with the cost function

$$J = \Psi(x(T)) + \int_0^T L(x(t), u(t))dt,$$

  only works with the fixed final time.

- We may implement the flexible time problem by time-rescaling functions.

Flexible final time

- From the original equation,

$$\begin{cases} \frac{d}{dt}x(t) = f(x(t), u(t)), \quad t \in [0, t_f], \end{cases}$$

we adopt a time-rescaling $T : [0, 1] \to [0, t_f]$, $t = T(s)$, where $T'(s) \geq c > 0$, $T'(s) < C$ for some c and C.

- Then, in terms of s, the dynamics of $\tilde{x}(s) = x(T(s))$ can be described by

$$\begin{aligned} \frac{d}{ds}\tilde{x}(s) = \frac{d}{ds}x(T(s)) = \frac{dT}{ds}\frac{d}{dt}x(t) \\ = T'(s)f(x(T(s)), u(T(s))) =: F(\tilde{x}(s), \tilde{u}(s), T(s)). \end{aligned}$$

Flexible final time

- One more question : How we can build the cost function:

$$\Psi(x(t_f)) = \tilde{\Psi}(\tilde{x}(1), T(1)),$$
$$L(x(t), u(t)) = \tilde{L}(\tilde{x}(s), \tilde{u}(s))T'(s).$$

- Therefore, we can obtain the optimal solution $\tilde{x}$, $\tilde{u}$ and T.

- In order to get the optimal solution for original equation, we need

$$x(t) = \tilde{x}(T^{-1}(t)) \quad \text{and} \quad u(t) = \tilde{u}(T^{-1}(t)).$$

- Moreover, if we want to minimize the final time t_f, then we may add $T'(s)$ to L, where $\int_0^1 T'(s)ds = t_f$.

A DyCon Toolbox code for the herding problem

```
N=1; M=1; syms t;

Y = sym('y',[8 1]); % States vectors for positions and velocities
ue = Y(1:2); ve = Y(3:4); ud = Y(5:6); vd = Y(7:8);
U = sym('u',[2 1]);
kappa = U(1); % Control function of the original problem
T = U(2); % Time-scaling from s to t

ur = ud-ue; % Relative position, driver - evader

f_e2 = @(x) (2./x);
f_d2 = @(x) -(-5.5./x+10./x.^2-2);
nu_e = 2.0;
nu_d = 2.0;
```

A DyCon Toolbox code for the herding problem

```
dot_ud = vd;
dot_ue = ve;
dot_vd = -f_d2(ur.'*ur)*ur - nu_d*vd + kappa * [-ur(2);ur(1)];
dot_ve = -f_e2(ur.'*ur)*ur - nu_e*ve;

F = [dot_ue;dot_ve;dot_ud;dot_vd]*T; % Multiply original velocities with time-scaling T(s).
Params = sym.empty;
numF = matlabFunction(F,'Var',{t,Y,U,Params});
Nt = 101; % Numerical time discretization
dt = 1/(Nt);
dynamics = ode(numF,Y,U,'FinalTime',1,'Nt',Nt);
% ud = (-3,0), ue = (0,0), and zero velocities initially.
dynamics.InitialCondition = [0;0;0;0;-3;0;0;0];
```

A DyCon Toolbox code for the herding problem

```
Psi  = 1000*(ue-uf).'*(ue-uf);
L    = 1*((kappa).^2)*T;

numPsi = matlabFunction(Psi,'Var',{t,Y});
numL = matlabFunction(L,'Var',{t,Y,U});

iP = Pontryagin(dynamics,numPsi,numL);

min_dt = 0.1;
iP.Constraints.Projector = @(Utline) [Utline(:,1),0.5*(Utline(:,end)-min_dt+abs(Utline(:,end)-min_dt))+min_dt];

U0_tline = [1.5662*ones(size(tline));5.1727*ones(size(tline))]';
```

A DyCon Toolbox code for the herding problem

```
tol = 1e-6;
GradientMethod(iP,U0_tline,'DescentAlgorithm',
    @ConjugateDescent, 'tol',tol,'tolU',tol,'tolJ',
    tol,'display','all','EachIter',20,'Graphs',true
    , 'GraphsFcn',{@graphs_init_GBR_flextime,
    @graphs_iter_GBR_flextime});
temp = iP.Solution.UOptimal;
```

One driver and one evader

We can observe that the optimal strategy is not an Off-Bang-Off control, but it shares the main idea: 'rotate and then drive'.

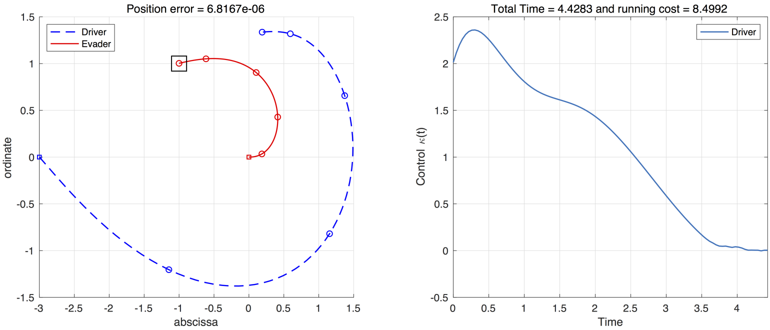


Figure: Diagrams for the optimal control leading to $\mathbf{u}_e(t_f) \simeq (-1, 1)$ when initially $\mathbf{u}_e^0 = (0, 0)$, $\mathbf{u}_d^0 = (-3, 0)$ and zero velocities.

Two drivers and one evader

This 'rotate and then drive' strategy also works with two drivers. In a similar initial data from the previous simulation, we can observe that two drivers act like one driver.

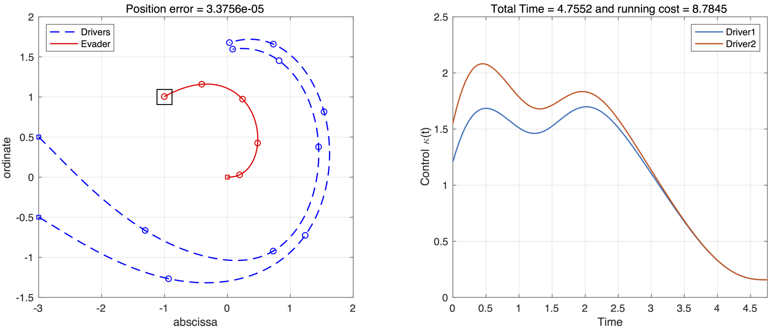


Figure: Diagrams for the control leading to $\mathbf{u}_{e1}(t_f) \simeq (-1, 1)$ when initially $\mathbf{u}_e^0 = (0, 0)$, $\mathbf{u}_{d1}^0 = (-3, 0.5)$, $\mathbf{u}_{d2}^0 = (-3, -0.5)$ and zero velocities.

Two drivers and one evader: Minimizing the driving time

It is not changed much even if we want to minimize the driving time,

$$J(\kappa(\cdot)) = \frac{1}{N}\sum_{i=1}^{N} |\mathbf{u}_{ei}(t_f) - \mathbf{u}_f|^2 + \frac{0.001}{M}\sum_{k=1}^{M}\int_0^{t_f} |\kappa_k(t)|^2 dt + 0.1 t_f.$$

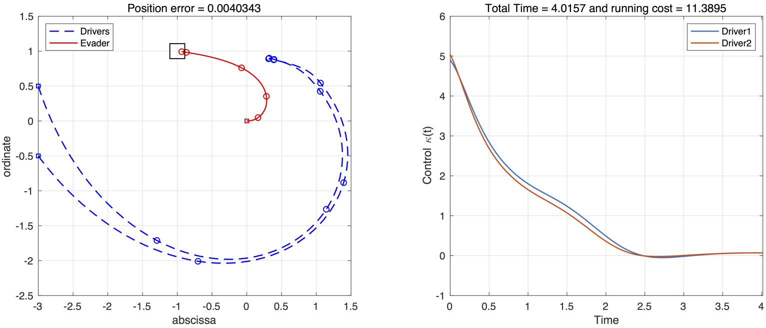


Figure: Diagrams for the control leading to $\mathbf{u}_{e1}(t_f) \simeq (-1, 1)$ when initially $\mathbf{u}_e^0 = (0, 0)$, $\mathbf{u}_{d1}^0 = (-3, 0.5)$, $\mathbf{u}_{d2}^0 = (-3, -0.5)$ and zero velocities.

The trajectories can be significantly different if initial positions are not well-ordered, in terms of the initial velocity of the evader. However, for any case, the drivers want the evader to get the right direction in a short time.

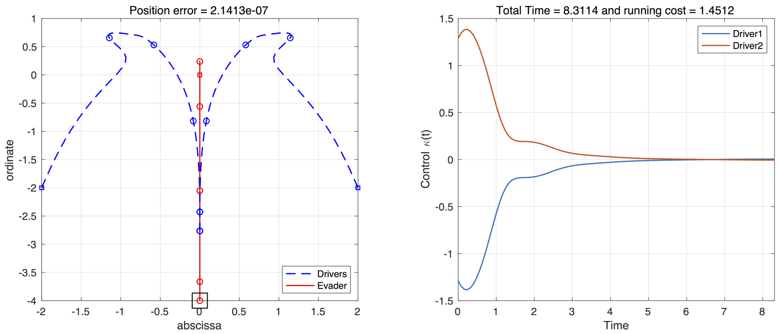


Figure: Diagrams for the control leading to $\mathbf{u}_{e1}(t_f) \simeq (0, -4)$ when initially $\mathbf{u}_e^0 = (0, 0)$, $\mathbf{u}_{d1}^0 = (-2, -2)$, $\mathbf{u}_{d2}^0 = (-2, 2)$ and zero velocities.

In the same way, the minimal time optimal strategy contains strong control functions and wants to decrease the relative position.

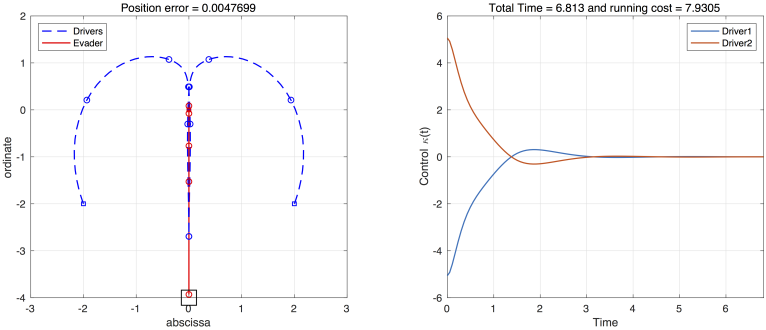


Figure: Diagrams for the control leading to $\mathbf{u}_{e1}(t_f) \simeq (0, -4)$ when initially $\mathbf{u}_e^0 = (0, 0)$, $\mathbf{u}_{d1}^0 = (-2, -2)$, $\mathbf{u}_{d2}^0 = (-2, 2)$ and zero velocities.