University of Wisconsin Milwaukee

UWM Digital Commons

Theses and Dissertations

August 2020

A Dynamic Programming Approach to Impulse Control of Brownian Motions

Robin Braun
University of Wisconsin-Milwaukee

Follow this and additional works at: https://dc.uwm.edu/etd

Part of the Mathematics Commons

Recommended Citation

Braun, Robin, "A Dynamic Programming Approach to Impulse Control of Brownian Motions" (2020). *Theses and Dissertations*. 2464.
https://dc.uwm.edu/etd/2464

This Thesis is brought to you for free and open access by UWM Digital Commons. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of UWM Digital Commons. For more information, please contact open-access@uwm.edu.

A DYNAMIC PROGRAMMING APPROACH TO IMPULSE CONTROL OF BROWNIAN MOTIONS

by

Robin Braun

A Thesis Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in Mathematics

at

The University of Wisconsin-Milwaukee

August 2020

ABSTRACT

A DYNAMIC PROGRAMMING APPROACH TO IMPULSE CONTROL OF BROWNIAN MOTIONS

by

Robin Braun

The University of Wisconsin-Milwaukee, 2020
Under the Supervision of Professor Chao Zhu

This thesis considers an impulse control problem of a standard Brownian motion under a discounted criterion, in which every intervention incurs a strictly positive cost. The value function and an optimal (τ_*, Y_*) policy are found using the dynamic programming principle together with the smooth pasting technique. The thesis also performs a sensitivity analysis by analyzing the limiting behaviors of the value function and the (τ_*, Y_*) policy when the fixed intervention cost converges to zero. It is demonstrated that the limits agree with the classic fuel follower problem.

The thesis next formulates and analyzes an N-player stochastic game of an impulse control problem under a discounted criterion. In the N-player stochastic game, each player controls an object. The objects are molded by an N-dimensional Brownian motion. A key aspect of the formulation is that each player aims to minimize her total impulse control cost and the total distance of her object to the moving center of the N objects. The interaction mandates the players to closely follow each other's movements. The Nash equilibrium is characterized and analyzed by a system of Hamilton-Jacobi-Bellman equations. The case when $N = 2$ is studied in detail.

© Copyright by Robin Braun, 2020
All Rights Reserved

Dedicated to my parents
who always kept me going when I showed any signs of quitting

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

DPP	dynamic programming principle
HJB	Hamilton–Jacobi–Bellman
NE	Nash equilibrium

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Professor Chao Zhu for giving me the chance to work on this project and all his support, helpful input and understanding during the work. I would also like to give special thanks to my thesis committee, Profs. Spade and Stockbridge. And of course, my thanks go to the math department for all their work and help during my stay in Milwaukee. Especially Prof. Willenbring for making this exchange program possible and Katie Wehrheim for helping me find an apartment when I arrived without a place to stay.

1 Introduction

In this thesis, our goal is to control a standard Brownian motion, where any intervention incurs a strictly positive cost. Doing this, means we have to select a sequence of separate intervention times and amounts, making the resulting stochastic problem an impulse control problem. Here, we want to minimize the total incurred cost over an infinite horizon, while considering a discounted criterion. In contrast to Helmes, Stockbridge and Zhu [1], which used a linear programming approach to solve the problem, we will use the dynamic programming principle (DPP) in combination with the smooth pasting technique to find the answer. This will give us an alternate approach to the problem that we can then try to extend to an N-dimensional case later on in the thesis.

1.1 1D Situation

Let W and $\{\mathcal{F}_t\}$ be a standard Brownian motion and its natural filtration. An impulse control policy is a pair of sequences $(\tau, Y) := \{(\tau_k, Y_k) : k \in \mathbb{N}\}$, where for each $k \in \mathbb{N}$, τ_k is an $\{\mathcal{F}_t\}$-stopping time and denotes the kth impulse time and Y_k is an $\mathcal{F}_{\tau_k}$-measurable variable indicating the kth impulse size.

Under such a policy, the controlled process is given by

$$\begin{aligned} X_t &:= x + W_t + \xi_t, \\ \xi_t &:= \sum_{k=1}^{\infty} I_{\{\tau_k \leq t\}} Y_k, \end{aligned} \tag{1.1}$$

where ξ is the accumulative jump amount up to time t. Note that $\Delta\xi_t := \xi_t - \xi_{t-} \neq 0$ if and only if an impulse τ_k occurs at time t and in that case $\Delta\xi_t = Y_k$.

Define the running cost h and the impulse cost c

$$h(x) = x^2, \qquad\qquad c(y,z) = k_1 + k_2|y - z|, \tag{1.2}$$

where k_1 is the fixed cost for an impulse, k_2 is the proportional cost and y and z are the pre- and post-jump locations. For a given impulse control policy (τ, Y) the cost functional is

$$J(\tau, Y; x) = \mathbb{E}_x \Bigg[\int_0^\infty e^{-\alpha s} \, h(X_s) ds \\ + \sum_{k=1}^{\infty} I_{\{\tau_k < \infty\}} e^{-\alpha \tau_k} \, c(X_{\tau_k -}, X_{\tau_k}) \Bigg], \tag{1.3}$$

where $x = X_{0-}$ is the initial position and $\alpha > 0$ is the discount factor. The first summand in (1.3) corresponds to the running cost and the second one to the control cost for an infinite horizon. In other words, the controller will have to balance the desire to keep the process near zero to keep the running cost low, against the desire to keep the number or sizes of interventions low as to not infer a great control cost.

The corresponding value-function is given by

$$V(x) = \inf_{(\tau, Y) \in \mathcal{U}} \{ J(\tau, Y; x) \}, \tag{1.4}$$

where $\mathcal{U}$ is the set of all admissible control policies (more on that in section 2).

The goal now is to find an impulse control pair (τ^*, Y^*) that minimizes the cost functional, i.e. for which $J(\tau^*, Y^*; x) = V(x)$ holds.

This problem has been solved in HSZ [1] using the linear programming approach. In the following section, we will also solve this problem using the DPP to give a different approach to the problem and then extend it to an N-dimensional version.

2 Analysis using the Dynamic Programming Principle

In this we will use the DPP to solve the problem stated in section 1.

Let's start by looking at the set $\mathcal{U}$ used in (1.4). One very important observation we can make, is that we don't need to include any policies (τ, Y) for which $J(\tau, Y; x) = \infty$ for some x, since our end goal is to minimize the cost.

2.1 Formal Derivation the HJB Equation

Before diving into finding the Hamilton–Jacobi–Bellman (HJB) equation, we'd like to make an important observation about the value-function and it's associated optimal policy: An optimal control policy, and with that V itself, should not be dependent on time, but only on the current position of the process. This is because optimizing the cost after some time t has past, is the same problem as optimizing from the start, just discounted by the factor $e^{-\alpha t}$.

With that, consider a policy $(\hat{\tau}, \hat{Y})$ that immediately jumps to an arbitrary position y and then continues optimally thereafter. We have

$$V(x) \leq J(\hat{\tau}, \hat{Y}; x) = c(x, y) + V(y) \qquad \forall y \in \mathbb{R}$$

Since the above equation holds true for any $y \in \mathbb{R}$ we have

$$V(x) - \mathcal{M}V(x) \leq 0 \qquad \forall x \in \mathbb{R}, \tag{2.1}$$

where $\mathcal{M}V(x) := \inf_{z \in \mathbb{R}} \{c(x, z) + V(z)\}$.

Next, consider a policy $(\tilde{\tau}, \tilde{Y})$ and its associated process $\tilde{X}$ that does nothing up to a time h and continues optimally from then on.

$$V(x) \leq J(\tilde{\tau}, \tilde{Y}; x) = \mathbb{E}_x \left[\int_0^h e^{-\alpha s} h(\tilde{X}_s) ds + e^{-\alpha h} V(\tilde{X}_h) \right] \tag{*}$$

Further assume that the value-function is smooth, so we can apply Itô's Formula to the process $e^{-\alpha t} V(\tilde{X}_t)$:

$$\begin{aligned} e^{-\alpha h} V(\tilde{X}_h) = V(x) &+ \int_0^h e^{-\alpha s} (-\alpha V + \frac{1}{2} V'')(\tilde{X}_s) ds \\ &+ \int_0^h e^{-\alpha s} V'(\tilde{X}_s) dW_s \end{aligned}$$

If we also assume that V' stays bounded (which is desirable, since we don't want the cost to explode for large $|x|$), the second summand above is a mean-zero martingale. Thus, by taking expectation on both sides, we get

$$\mathbb{E}_x \left[e^{-\alpha h} V(\tilde{X}_h) \right] = V(x) + \mathbb{E}_x \left[\int_0^h e^{-\alpha s} (-\alpha V + \frac{1}{2} V'')(\tilde{X}_s) ds \right].$$

Plugging this into (*) yields

$$\begin{aligned} V(x) &\leq \mathbb{E}_x \left[\int_0^h e^{-\alpha s} h(\tilde{X}_s) ds \right] + V(x) + \mathbb{E}_x \left[\int_0^h e^{-\alpha s} (-\alpha V + \frac{1}{2} V'')(\tilde{X}_s) ds \right] \\ 0 &\leq \mathbb{E}_x \left[\int_0^h e^{-\alpha s} (-\alpha V + \frac{1}{2} V'' + h)(\tilde{X}_s) ds \right]. \end{aligned}$$

Dividing by h and letting $h \to 0$ gives us

$$\begin{aligned} 0 &\leq e^{-\alpha \cdot 0} (-\alpha V + \frac{1}{2} V'' + h)(\tilde{X}_0) \\ 0 &\geq (\alpha V - \frac{1}{2} V'' - h)(x) \qquad \forall x \in \mathbb{R}. \end{aligned} \tag{2.2}$$

Now, if we consider the optimal policy (τ^*, Y^*) at an initial position $X_{0-} = x$ at time zero, it only has two options. Either it will immediately push the process, or it will idle. In the first case we get that the process jumps to some $z \in \mathbb{R}$, so

$$V(x) = J(\tau^*, Y^*; x) = c(x, z) + V(z).$$

Since our policy is optimal, we also get

$$V(x) = J(\tau^*, Y^*; x) \leq c(x, y) + V(y) \forall y \neq z,$$

giving us in total:

$$V(x) - \mathcal{M}V(x) = 0$$

In the second case, for $h > 0$ small enough, we get

$$V(x) = J(\tau^*, Y^*; x) = \mathbb{E}_x \left[\int_0^h e^{-\alpha s} h(X_s^*) ds + e^{-\alpha h} V(X_h^*) \right].$$

Following the same steps as for $(\tilde{\tau}, \tilde{Y})$, we get

$$0 = (\alpha V - \frac{1}{2}V'' - h)(x).$$

Putting our two cases together, means the value-function must satisfy

$$\max \left\{ \alpha V(x) - \frac{1}{2}V''(x) - h(x), V(x) - \mathcal{M}V(x) \right\} = 0. \tag{2.3}$$

2.2 Finding the Optimal Strategy

Now that we know the HJB equation for V, let's think about how an optimal strategy could look like. We already know that it shouldn't be dependent on time, i.e. it will always act the same for the same position. This means we can split our space in two regions: The Action Region $\mathcal{A}$ and Continuation Region or Waiting Region $\mathcal{W}$ (We push when the process is in $\mathcal{A}$ and idle in $\mathcal{W}$):

$$\mathcal{A} := \{x \in \mathbb{R} : \Delta\xi(x) \neq 0\}, \qquad\qquad \mathcal{W} := \mathcal{A}^C \tag{2.4}$$

Pushes should always move the process out of $\mathcal{A}$, since otherwise we would push again immediately afterwards, incurring the fixed cost twice.

Another observation is that our strategy should be symmetrical, meaning if it pushes to z from a position y, it should push to $-z$ from $-y$. This is because the running cost $h(x) = x^2$ is even and the push cost doesn't favor one direction over the other. Also, since h is strictly increasing for growing $|x|$, there shouldn't be a part of $\mathcal{W}$ farther away from 0 than $\mathcal{A}$, i.e.:

$$\forall x \in \mathcal{W}, \forall y \in \mathcal{A} : |x| \leq |y|$$

So we should expect $\mathcal{W}$ to be some area bounded around zero and $\mathcal{A}$ to be the rest of $\mathbb{R}$. This implies there exists some $y_* > 0$ (to be determined) s.t.:

$$\mathcal{W} = \{x \in \mathbb{R} : -y_* \leq x \leq y_*\}$$

The last remaining question is how much we should push if x is in $\mathcal{A}$. Let $x > y_*$ (so $x \in \mathcal{A}$). Since we want the optimal strategy, we should expect there to be some $z_* \in \mathcal{W}$ (to be determined) that is the optimal place to push to. In that case z_* should also be the optimal place for all $x > y_*$ since the push cost is linearly proportional to the push distance (factor k_2).

So, all in all, the strategy should idle in $\mathcal{W}$ and always push to z_* for $x > y_*$ or, because of the symmetry, to $-z_*$ for $x < -y_*$ (See Figure 1). So in mathematical terms, our strategy (τ^*, Y^*) is defined the following way:

$$\begin{aligned} \tau_0^* &:= \inf\{t > 0 : |X_t| > y_*\} \\ \tau_k^* &:= \inf\{t > \tau_{k-1}^* : |X_t| > y_*\} \quad \forall k \in \mathbb{N}_> \\ Y_k^* &:= sgn(X_{\tau_n^*-}) \cdot z_* - X_{\tau_n^*-} \qquad \forall k \in \mathbb{N}_\geq \end{aligned} \tag{2.5}$$

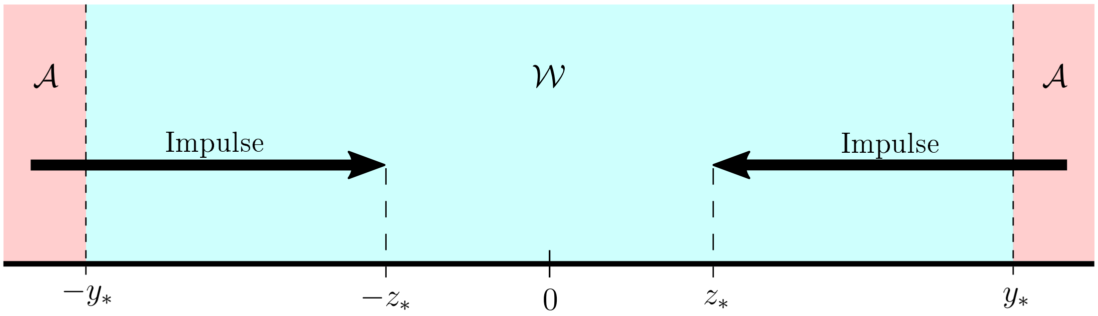


Figure 1: 1D Impulse control strategy

Now we need to determine the associated value $u(x) := J(\tau^*, Y^*; x)$ to the optimal strategy, and with it y_* and z_*.

From the symmetry we know that $u(x) = u(-x)$ if $x < 0$. For $x > y_*$, we always push to z_*, so u looks like the following:

$$u(x) = u(z_*) + c(x, z_*)$$

For $0 \leq x \leq y_*$ our policy idles, so we want $u(x)$ to fulfill $\alpha u(x) - \frac{1}{2}u''(x) - h(x) = 0$ (see 2.3).

To find that, we first solve the homogeneous part of the differential equation.

$$\alpha u_h(x) - \frac{1}{2}u_h''(x) = 0 \Rightarrow \alpha - \frac{1}{2}\lambda^2 = 0 \Rightarrow \lambda = \pm\sqrt{2\alpha} =: \pm\rho$$

Which gives us the homogeneous solution

$$u_h(x) = A_1 e^{-\rho x} + A_2 e^{\rho x} \qquad A_1, A_2 \text{ constants}$$

Since $h(x) = x^2$, we can find the particular solution by assuming it is a quadratic function, $u_p(x) = ax^2 + bx + c$:

$$\alpha u_p(x) = \frac{1}{2}u_p''(x) + x^2 \Rightarrow (ax^2 + bx + c) = \frac{1}{2}2a + x^2 \Rightarrow a = \frac{1}{\alpha}; b = 0; c = \frac{a}{\alpha} = \frac{1}{\alpha^2}$$

Which gives us the particular solution

$$u_p(x) = \frac{1}{\alpha}x^2 + \frac{1}{\alpha^2} = \frac{\alpha x^2 + 1}{\alpha^2}$$

Remark 2.1

A more general approach for cost functions $c(x)$ other than x^2 is to use the Zero-control (do nothing) to find the particular solution (See Guo and Xu, 2019 [2]):

$$\begin{aligned} u_p(x) &= \mathbb{E}\left[\int_0^\infty e^{-\alpha t} c(X_t) dt\right] = \mathbb{E}\left[\int_0^\infty e^{-\alpha t} h(x + W_t) dt\right] \\ &= \int_0^\infty e^{-\alpha t} \int_{\mathbb{R}} c(x+y) \frac{1}{\sqrt{2\pi t}} e^{-\frac{y^2}{2t}} \, dy \, dt \\ &= \int_{\mathbb{R}} c(x+y) \int_0^\infty e^{-\alpha t} \frac{1}{\sqrt{2\pi t}} e^{-\frac{y^2}{2t}} \, dt \, dy \\ &= \int_{\mathbb{R}} c(x+y) \, \frac{1}{\sqrt{2\alpha}} e^{-\sqrt{2\alpha}|y|} dy \end{aligned}$$

All in all, this gives us $u(x) = u_h(x) + u_p(x) = A_1 e^{-\rho x} + A_2 e^{\rho x} + \frac{\alpha x^2+1}{\alpha^2}$
From the symmetry we get:

$$u(x) = A_1 e^{-\rho x} + A_2 e^{\rho x} + \frac{\alpha x^2+1}{\alpha^2} = A_1 e^{\rho x} + A_2 e^{-\rho x} + \frac{\alpha x^2+1}{\alpha^2} = u(-x)$$

So A_1 has to be equal to A_2. Defining $A := 2A_1$ lets us rewrite $u(x)$ the following way:

$$u(x) = A\cosh(\rho x) + \frac{\alpha x^2+1}{\alpha^2} \qquad \forall 0 \leq x < y_*$$

So $u(x)$ is defined as follows:

$$u(x) = \begin{cases} A\cosh(\rho x) + \frac{\alpha x^2+1}{\alpha^2} & 0 \leq x \leq y_* \\ u(z_*) + c(x, z_*) & x > y_* \\ u(-x) & x < 0 \end{cases} \tag{2.6}$$

Notice that per this definition $u'(x)$ is bounded, which was one of our assumptions for the value-function. Another assumption was for V to be smooth, so we want u to fulfill this also. For that we use the Smooth Pasting technique, i.e. set u and its derivative equal on the boundaries:

$$\begin{aligned} u(y_*-) &= A\cosh(\rho y_*) + \frac{\alpha y_*^2+1}{\alpha^2} \\ &= A\cosh(\rho z_*) + \frac{\alpha z_*^2+1}{\alpha^2} + c(y_*, z_*) = u(y_*+) \\ u'(y_*-) &= \rho A\sinh(\rho y_*) + \frac{2y_*}{\alpha} = k_2 = u'(y_*+) \end{aligned} \tag{2.7}$$

We now have two equations, but need to determine three variables (A, y_* and z_*), so we need one more constraint. We get that from the fact that we push to z_*. But if we are doing a push, it should be the optimal one, so the following holds for an $x > y_*$:

$$\frac{\partial}{\partial z}\left[u(z_*) + c(x, z_*)\right] = 0 \Leftrightarrow u'(z_*) - k_2 = 0 \Leftrightarrow \rho A \sinh(\rho z_*) + \frac{2z_*}{\alpha} = k_2 \tag{2.8}$$

Remark 2.2

From the last two constraints we can already make an important observation: Since both $2x/\alpha$ and $\rho \sinh(x)$ are strictly increasing functions, the only way to get a $y_* \neq z_*$ is when $A < 0$.

If we use the first constraint of (2.7) and solve for A, we get:

$$\begin{aligned} A\cosh(\rho y_*) - A\cosh(\rho z_*) &= \frac{\alpha z_*^2 + 1}{\alpha^2} - \frac{\alpha y_*^2 + 1}{\alpha^2} + c(y_*, z_*) \\ A &= \frac{c(y_*, z_*) + \frac{z_*^2 - y_*^2}{\alpha}}{\cosh(\rho y_*) - \cosh(\rho z_*)} \qquad (\equiv A(z_*, y_*)) \end{aligned}$$

Rearranging the other two constraints a little, we get:

$$\begin{aligned} \alpha \cdot \rho \sinh(\rho y_*) \cdot A &= k_2\alpha - 2y_* \qquad \left(\Rightarrow y_* > \frac{k_2\alpha}{2}\right) \\ \alpha \cdot \rho \sinh(\rho z_*) \cdot A &= k_2\alpha - 2z_* \qquad \left(\Rightarrow z_* > \frac{k_2\alpha}{2}\right) \end{aligned}$$

So:

$$A = \frac{c(y_*, z_*) + \frac{z_*^2 - y_*^2}{\alpha}}{\cosh(\rho y_*) - \cosh(\rho z_*)} = \frac{k_2\alpha - 2y_*}{\alpha \cdot \rho \sinh(\rho y_*)} = \frac{k_2\alpha - 2z_*}{\alpha \cdot \rho \sinh(\rho z_*)} \tag{2.9}$$

These equations have a unique solution and an analysis of the function $A(z, y)$ shows that it has a unique minimum less than zero at (z_*, y_*) (see HSZ VI-VII [1]). Unfortunately there is no straightforward analytic expression for y_* and z_*. It is however relatively straightforward to get them numerically (see section 2.4).

In the following figure, we can see a plot of u with specific parameters. Notice that the tangents at y_* and z_* have the same angle ϕ (from $u'(z_*) = u'(y_*)$) and the function continues linearly for $|x| > y_*$.

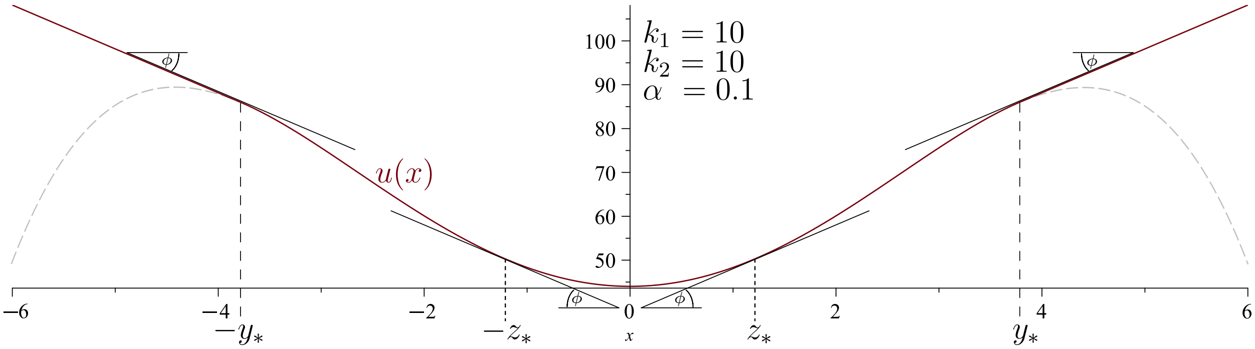


Figure 2: Sample plot of the candidate function $u(x)$

Using the script to find y_* and z_*, we can also take a look at a sample path of the controlled and uncontrolled processes.

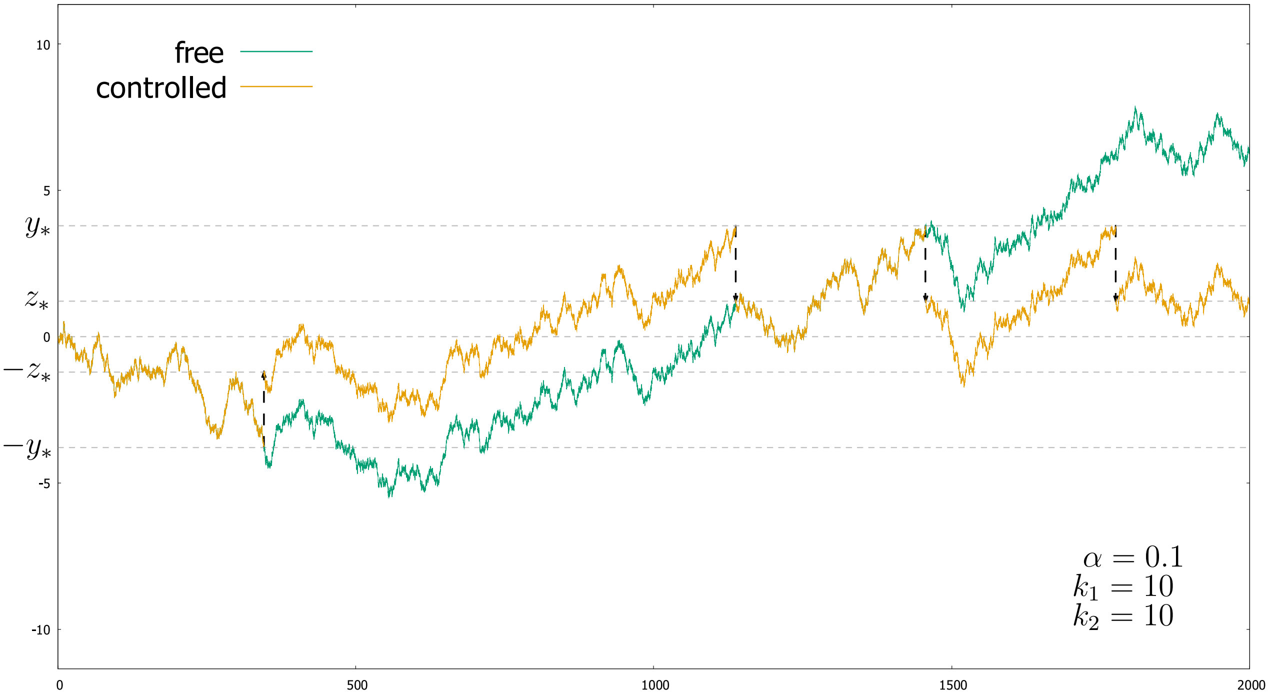


Figure 3: Sample paths of the uncontrolled and optimally controlled processes.

2.3 Verification

Proposition 2.3

$u \in C^1(\mathbb{R}) \cap C^2(\mathbb{R} \setminus \{-y_*, y_*\})$ as given in (2.6) and satisfying (2.9) is a solution to the HJB equation given in (2.3).

Proof. Per construction, we know that on $\mathcal{W}$, $(\alpha u - \frac{1}{2}u'' - h)(x) \leq 0$ is fulfilled. It remains to check whether $u(x) - \mathcal{M}u(x) \leq 0$. From the symmetry it's enough to consider $0 \leq x \leq y_*$. Since u is strictly increasing on $\mathbb{R}_+$ we know that for a $z > x$, $u(z) > u(x)$, so the above inequality is trivially fulfilled. So the interesting case is for a $0 \leq z \leq x$:

$$\begin{aligned} u(x) - u(z) - c(x,z) &\overset{?}{\leq} 0 \\ A\cosh(\rho x) + \frac{\alpha x^2 + 1}{\alpha} - A\cosh(\rho z) - \frac{\alpha z^2 + 1}{\alpha} &\overset{?}{\leq} k_1 + k_2(x - z) \\ A\left(\cosh(\rho x) - \cosh(\rho z)\right) &\overset{?}{\leq} k_1 + k_2(x-z) + \frac{z^2 - x^2}{\alpha} \\ A &\leq \frac{k_1 + k_2(x-z) + \frac{z^2 - x^2}{\alpha}}{\cosh(\rho x) - \cosh(\rho z)} \end{aligned}$$

This inequality holds, since per definition of u, A is the minimum of the right expression. So in $\mathcal{W}$ the HJB equation is satisfied. What's left is to check $\mathcal{A}$, i.e. $x > y_*$:

Per construction we know that $u(x) - \mathcal{M}u(x) = 0$. So what's left is to show the following:

$$\begin{aligned} \alpha w(x) - \frac{1}{2}w''(x) - h(x) &\overset{?}{\leq} 0 \qquad (w''(x) = 0) \\ \alpha w(x) - x^2 &\overset{?}{\leq} 0 \end{aligned}$$

$x > y_*$ implies that

$$\frac{d}{dx}\left[\alpha w(x) - x^2\right] = k_2\alpha - 2x < k_2\alpha - 2y_* = \alpha \cdot \rho\sinh(\rho y_*) \cdot A < 0,$$

So it is enough to show that $\alpha u(y_*) - y_*^2 \leq 0$. But $y_* \in \mathcal{W}$, so the following holds:

$$\alpha u(y_*) - y_*^2 = \frac{1}{2}u''(y_*)$$

From the definition of u we can make three important observations. Firstly u' is a concave function on $\mathbb{R}_+$, secondly $u'(0) = 0$ and $u'(z_*) = u'(y_*) = k_2 > 0$, which gives us thirdly that u' has its absolute maximum on $\mathbb{R}_+$ somewhere between z_* and y_*. This implies that $u''(z_*) \geq 0$ and $u''(y_*) \leq 0$. So in particular $\frac{1}{2}u''(y_*) \leq 0$.

So u satisfies the HJB equation. □

Proposition 2.4

For u defined as in (2.3) and (τ^*, Y^*) defined as in (2.5): $u(x) = J(\tau^*, Y^*; x)$.

Proof. Since u is smooth, we can apply Itô's Formula on the process $e^{-\alpha t}u(X_t)$:

$$\begin{aligned} e^{-\alpha t}u(X_t) = u(x) &+ \int_0^t e^{-\alpha s}(-\alpha u + \frac{1}{2}u'')(X_s)ds \\ &+ \int_0^t e^{-\alpha s}u'(X_s)dW_s \\ &+ \sum_{k=0}^{\infty} I_{\{\tau_k^* \leq t\}} e^{-\alpha \tau_k^*}\left[u(X_{\tau_k^*}) - u(X_{\tau_k^*-})\right] \end{aligned} \tag{2.10}$$

Applying expectation, rearranging a bit and adding and subtracting $h(X_t)$ we get

$$\begin{aligned} \mathbb{E}_x[u(x)] = \mathbb{E}_x &\left[e^{-\alpha t}u(X_t)\right] \\ &+ \mathbb{E}_x\left[\int_0^t e^{-\alpha s}(\alpha u - \frac{1}{2}u'' - h)(X_s)ds\right] \\ &+ \mathbb{E}_x\left[\int_0^t e^{-\alpha s}h(X_s)ds\right] \\ &- \mathbb{E}_x\left[\int_0^t e^{-\alpha s}u'(X_s)dW_s\right] \\ &+ \mathbb{E}_x\left[\sum_{k=0}^{\infty} I_{\{\tau_k^* \leq t\}} e^{-\alpha \tau_k^*}\left(u(X_{\tau_k^*-}) - u(X_{\tau_k^*})\right)\right] \end{aligned} \tag{2.11}$$

Remember that for this policy we always immediately push into $\mathcal{W}$ for any $x \in \mathcal{A}$. This means for one that X_t stays bounded, so the first term approaches zero as t approaches infinity. It also implies that the set $\{t \geq 0 : X_t \in \mathcal{A}\}$ has Lebesgue measure zero. Furthermore, from the construction we know that for any $x \in \mathcal{W}$ the following holds: $u(x) - \frac{1}{2}u''(x) - h(x) = 0$. Which means the second term is equal to zero. Per construction, u' is bounded, so the fourth term is equal to zero. Also, for this control $|X_{\tau_k^*-}| \geq y_*$ and $|X_{\tau_k^*}| = z_*$, so

$$\begin{aligned} & u(X_{\tau_k^*-}) = u(\operatorname{sgn}(X_{\tau_k^*-})z_*) + c(X_{\tau_k^*-}, \operatorname{sgn}(X_{\tau_k^*-})z_*) = u(X_{\tau_k^*}) + c(X_{\tau_k^*-}, X_{\tau_k^*}) \\ \Rightarrow\ & u(X_{\tau_k^*-}) - u(X_{\tau_k^*}) = c(X_{\tau_k^*-}, X_{\tau_k^*}) \end{aligned}$$

This gives us first that

$$\begin{aligned} u(x) &= \mathbb{E}_x\left[e^{-\alpha t}u(X_t)\right] \\ &+ \mathbb{E}_x\left[\int_0^t e^{-\alpha s}h(X_s)ds\right] \\ &+ \mathbb{E}_x\left[\sum_{k=0}^{\infty} I_{\{\tau_k^* \leq t\}} e^{-\alpha\tau_k^*} c(X_{\tau_k^*-}, X_{\tau_k^*})\right] \end{aligned}$$

and then, by applying the monotone convergence theorem

$$\begin{aligned} u(x) &\overset{t\to\infty}{=} \mathbb{E}_x\left[\int_0^{\infty} e^{-\alpha s}h(X_s)ds\right] + \mathbb{E}_x\left[\sum_{k=0}^{\infty} I_{\{\tau_k^* < \infty\}} e^{-\alpha\tau_k^*} c(X_{\tau_k^*-}, X_{\tau_k^*})\right] \\ &= J(\tau^*, Y^*; x) \end{aligned} \tag{2.12}$$

This finishes the proof. □

Theorem 2.1 (Verification Theorem)**.** *Suppose (τ^*, Y^*) is an admissible policy and the corresponding value $u(x) := J(\tau^*, Y^*; x)$ with*

(i) *$u(x) \in C^2(\mathbb{R})$ and satisfies (2.3),*

(ii) *$u'(x)$ is bounded,*

(iii) *for any (τ, Y) and its controlled dynamic X_t the transversality condition holds:*

$$\limsup_{t\to\infty} \mathbb{E}_x \left[e^{-\alpha t} u(X_t)\right] \leq 0.$$

Then

$$u(x) \leq J(\tau, Y; x) \quad \forall (\tau, Y) \text{ admissible policies,}$$

i.e. $V(x) = u(x)$.

Proof. Under a policy (τ, Y), we have

$$X_t = x + W_t + \sum_{k=1}^{\infty} I_{\{\tau_k \leq t\}} Y_k$$

We now apply Itô's Formula to the process $e^{-\alpha t} u(X_t)$ as in (2.10) and (2.11) to obtain

$$\begin{aligned} u(x) &= \mathbb{E}_x \left[e^{-\alpha t} u(X_t)\right] \\ &+ \mathbb{E}_x \left[\int_0^t e^{-\alpha s} (\alpha u - \frac{1}{2} u'' - h)(X_s) ds\right] \\ &+ \mathbb{E}_x \left[\int_0^t e^{-\alpha s} h(X_s) ds\right] \\ &- \mathbb{E}_x \left[\int_0^t e^{-\alpha s} u'(X_s) dW_s\right] \\ &+ \mathbb{E}_x \left[\sum_{k=0}^{\infty} I_{\{\tau_k \leq t\}} e^{-\alpha \tau_k} \left(u(X_{\tau_k -}) - u(X_{\tau_k})\right)\right] \end{aligned} \tag{2.13}$$

Letting $t \to \infty$, the first term is 0 because of the transversality condition. Since $u(x)$ satisfies (2.3), the second term is less than or equal to 0 and $u(X_{\tau_k-}) - u(X_{\tau_k}) \leq c(X_{\tau_k-}, X_{\tau_k})$ for the last term. Since $u'(x)$ is bounded, the fourth term is equal to 0. Which gives us in total

$$\begin{aligned} u(x) &\overset{t\to\infty}{\leq} \mathbb{E}_x\left[\int_0^\infty e^{-\alpha s} h(X_s) ds\right] + \mathbb{E}_x\left[\sum_{k=0}^\infty I_{\{\tau_k<\infty\}} e^{-\alpha\tau_k} c(X_{\tau_k-}, X_{\tau_k})\right] \\ &= J(\tau, Y; x) \end{aligned} \tag{2.14}$$

This ends the proof. □

As we can see, our u fits the Verification Theorem if we can show the transversality condition. Let (τ, Y) be an admissible policy and X_t the corresponding controlled process. Suppose further, the transversality condition does not hold, i.e.

$$\liminf_{t\to\infty} \mathbb{E}_x\left[e^{-\alpha t} u(X_t)\right] > K \text{ for some } K > 0.$$

This implies that X_t is unbounded, so

$$\lim_{t\to\infty} \mathbb{E}_x\left[e^{-\alpha t} u(X_t)\right] = \lim_{t\to\infty} \mathbb{E}_x\left[e^{-\alpha t} u(X_t) I_{\{|X_t|\geq y_*\}}\right].$$

The linearity of u on $\{x : |x| \geq y_*\}$ implies that

$$\begin{aligned} \lim_{t\to\infty} \mathbb{E}_x\left[e^{-\alpha t} u(|X_t|) I_{\{|X_t|\geq y_*\}}\right] &> K \\ \lim_{t\to\infty} \mathbb{E}_x\left[e^{-\alpha t}\left(u(z_*) + k_1 + k_2(|X_t| - z_*)\right) I_{\{|X_t|\geq y_*\}}\right] &> K \\ \lim_{t\to\infty} e^{-\alpha t}(u(z_*) + k_1 - k_2 z_*) + k_2 e^{-\alpha t} \lim_{t\to\infty} \mathbb{E}_x\left[|X_t| I_{\{|X_t|\geq y_*\}}\right] &> K \\ \lim_{t\to\infty} \mathbb{E}_x\left[|X_t| I_{\{|X_t|\geq y_*\}}\right] &> \lim_{t\to\infty} e^{\alpha t} k_2 \cdot K \end{aligned}$$

So $\mathbb{E}_x\left[|X_t| I_{\{|X_t|\geq y_*\}}\right]$ is asymptotically bounded below by $e^{\alpha t}\tilde{K}$. Jensen's inequality tells us

$\mathbb{E}_x\left[|X_t|\right]^2 \leq \mathbb{E}_x\left[X_t^2\right]$. Combined with the above for an $\epsilon > 0$ and large enough t we get

$$\mathbb{E}_x\left[X_t^2\right] \geq \mathbb{E}_x\left[|X_t| I_{\{|X_t| \geq y_*\}}\right]^2 \geq (\tilde{K}e^{\alpha t})^2 - \epsilon,$$

which implies $J(\tau, Y; x) = \infty$, which means (τ, Y) wasn't an admissible control to begin with.

To summarize, we get the following.

Theorem 2.2. *Let u be defined by (2.3) and (τ^*, Y^*) defined as in (2.5). Then u is the value-function, i.e.*

$$J(\tau, Y; x) \leq u(x) = J(\tau^*, Y^*; x) \qquad \forall x \in \mathbb{R}, (\tau, Y) \in \mathcal{U}$$

Meaning we have found the value-function and the optimal control policy.

2.4 Numerical Solution for y_* and z_*

Unfortunately, we can not find a simple analytic expression for z_* and y_*, but finding them numerically is relatively straight-forward. Remember, (z_*, y_*) is the pair that minimizes

$$A(z,y) = \frac{k_1 + k_2(y_* - z_*) + \frac{z_*^2 - y_*^2}{\alpha}}{\cosh(\rho y_*) - \cosh(\rho z_*)}$$

on the domain $0 < z < y$. Since $u'(z_*) = u'(y_*)$, we also get

$$\frac{2y_* - k_2\alpha}{\sinh(\rho y_*)} = \frac{2z_* - k_2\alpha}{\sinh(\rho z_*)}.$$

If we take a look at the function

$$f : \left(\frac{k_2\alpha}{2}, \infty\right) \ni x \to \frac{2x - k_2\alpha}{\sinh(\rho x)} \in \mathbb{R}_+,$$

we see that it has a maximum at $(\hat{x}, \hat{t})$, is strictly increasing on $\left(\frac{k_2\alpha}{2}, \hat{x}\right)$ and strictly decreasing on $(\hat{x}, \infty)$.

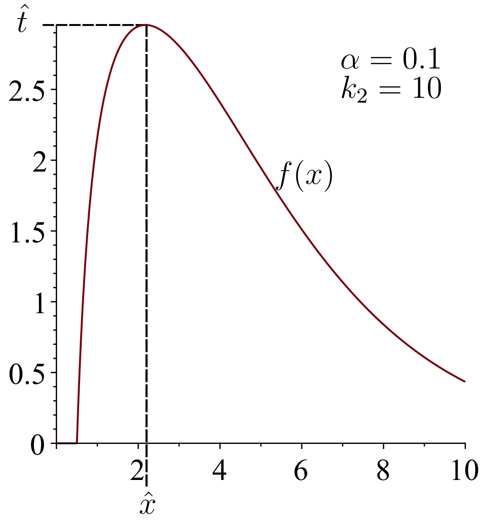


Figure 4: Sample plot of $f(x)$

So for each section f has an inverse function. This allows us to define a function that gives us all level sets (z, y):

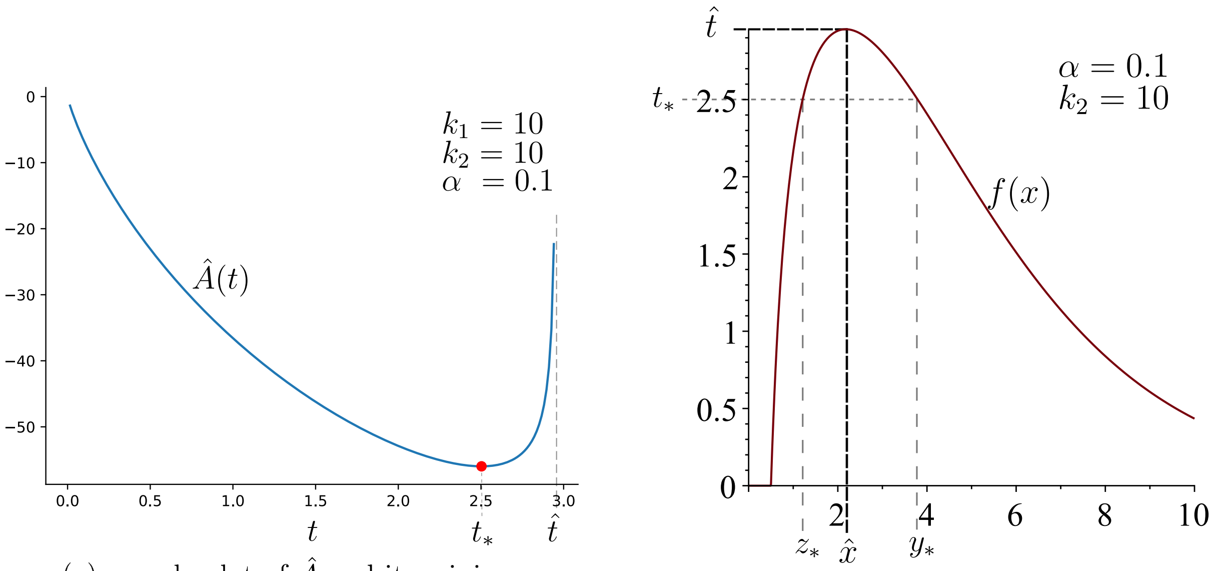
$$p : (0, \hat{t}) \ni t \to \begin{pmatrix} f\big|_{\left(\frac{k_2\alpha}{2}, \hat{x}\right)}^{-1}(t) \\ f\big|_{(\hat{x},\infty)}^{-1}(t) \end{pmatrix} \in \left(\frac{k_2\alpha}{2}, \hat{x}\right) \times (\hat{x}, \infty)$$

Using this, we get a new function, only dependent on a single variable that we need to minimize:

$$\hat{A} : (0, \hat{t}) \ni t \to A(p(t)) \in \mathbb{R}$$

So if we define $t_* := \arg\min_{t\in(0,\hat{t})} \left\{\hat{A}(t)\right\}$, we get that

$$p(t_*) = (z_*, y_*)^T, \qquad \hat{A}(t_*) = A$$



(a) sample plot of $\hat{A}$ and its minimum

(b) t_*, z_* and y_* marked in a plot of $f(x)$

Figure 5: Plots of the numerical Solution

Based on this we created a python script to find (z_*, y_*) and A given α, k_1 and k_2 (See Appendix Python-Code 1-D).

2.5 Comparison with the Fuel Follower Problem

Guo and Xu [2] discuss a very similar problem in their paper: The fuel follower problem. This problem is very similar to our impulse control problem, with the only difference, that there is no fixed cost k_1. For their discussion, they also used a fixed push-cost of 1, which would mean $k_2 = 1$ for our problem.

The optimal policy they found was to always apply the minimal push for the process to stay within $[-c, c]$ for a $c > 0$ and the associated value-function is given by

$$v_f(x) = \begin{cases} -\frac{p_1''(c)\cosh(x\sqrt{2\alpha})}{2\alpha\cosh(c\sqrt{2\alpha})} + p_1(x) & 0 \leq x \leq c \\ v_f(c) + (x - c) & x \geq c \\ v_f(-x) & x < 0. \end{cases} \tag{2.15}$$

Where p_1 is the cost of the "Zero-Control". Using the same running cost function as in this thesis ($h(x) = x^2$) will give us $p_1(x) = u_p(x) = \frac{\alpha x^2+1}{\alpha^2}$.

We can see that both the policy and value-function are very similar to our result. The key difference is that Guo and Xu push to the border of the waiting region (to c), whereas we actually push to a point inside it (to z_*).

An interesting question that that we can now ask ourselves is what happens when we let the fixed cost k_1 go to zero, i.e. if we make our problem more and more similar to the fuel follower problem. Intuition tells us we should expect the same result, meaning we should move to a strategy that pushes to the border of the waiting instead of inside it, or in other words, we should expect to see $z_* \to y_*$ as $k_1 \to 0$.

One thing that illustrates this quite well is if we take a look at the following function and its derivative:

$$G(x) := \frac{\alpha x^2 + 1}{\alpha^2} + A\cosh(\rho x) - k_2 \cdot x \;\Rightarrow\; G'(x) := \frac{2x}{\alpha} + A \cdot \rho \sinh(\rho x) - k_2$$

This allows us to express the constraints for u in (2.7) and (2.8) in terms of G:

$$G'(y_*) = 0, \qquad G'(z_*) = 0, \qquad G(y_*) = G(z_*) + k_1$$
$$\Rightarrow\; k_1 = \int_{z_*}^{y_*} G'(x)\; dx$$

Since we can now express k_1 as the area of G' (which is strictly concave) between z_* and y_*, it is obvious that $k_1 \to 0$ has to imply that $z_* \to y_*$ (see Figure 6).

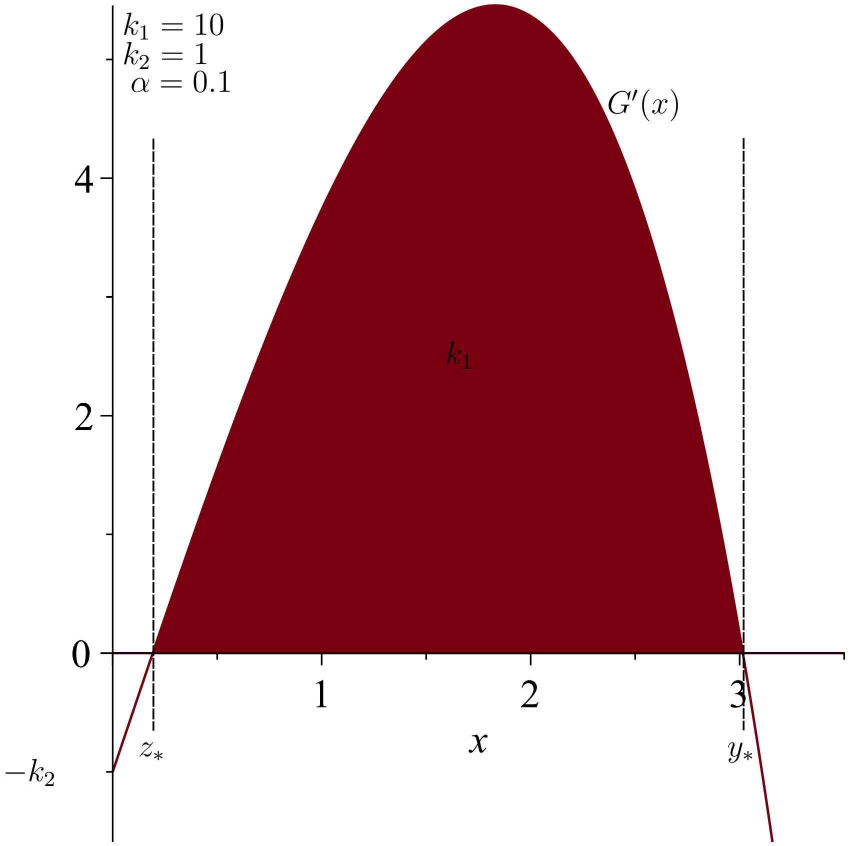


Figure 6: k_1 expressed as an area of G'

Since we now know that $z_* \to y_*$, this has to happen at the maximum (let's call it c to go with Guo and Xu's [2] convention) of $f(x) = \frac{2x-k_2\alpha}{\sinh(\rho x)}$.

$$f'(c) = \frac{2\sinh(\rho c) - (2c - k_2\alpha)\rho\cosh(\rho c)}{(\sinh(\rho c))^2} = 0$$

$$2\sinh(\rho c) = (2c - k_2\alpha)\rho\cosh(\rho c)$$

$$\frac{\sinh(\rho c)}{\rho\cosh(\rho c)} = \frac{\tanh(\rho c)}{\rho} = \frac{2c - k_2\alpha}{2} \qquad \left(\overset{k_2=1}{=} \frac{2c-\alpha}{2}\right)$$

In GX [2] c is the solution to $\frac{1}{\sqrt{2\alpha}}\tanh(c\sqrt{2\alpha}) = \frac{p_1'(c)-1}{p_1''(c)}$. Together with $\rho = \sqrt{2\alpha}$ and $p_1(x) = \frac{\alpha x^2+1}{\alpha^2}$, we get

$$\frac{1}{\rho}\tanh(\rho c) = \frac{1}{\sqrt{2\alpha}}\tanh(c\sqrt{2\alpha}) = \frac{p_1'(c)-1}{p_1''(c)} = \frac{\frac{2c}{\alpha}-1}{\frac{2}{\alpha}} = \frac{2c-\alpha}{2},$$

meaning the c's are the same in the two problems. Left to check is, if the value-functions also agree. If we compare our value function u (2.6) with their value-function v_f (2.15), we can see that they are already similar in the last two cases, so left is to check if $u(x) = v_f(x)$ for any $0 \leq x \leq c$. In our problem we have $u(x) = A\cosh(\rho x) + u_p(x)$, $A = \frac{k_2\alpha - 2c}{\alpha\cdot\rho\sinh(\rho c)}$ and from before $\sinh(\rho c) = \rho\cosh(\rho c)\frac{2c-k_2\alpha}{2}$, so

$$\begin{aligned} u(x) &= \frac{k_2\alpha - 2c}{\alpha\cdot\rho\rho\cosh(\rho c)\frac{2c-k_2\alpha}{2}}\cosh(\rho x) + u_p(x) \\ &= -\frac{2}{\alpha 2\alpha\cosh(\rho c)}\cosh(\rho x) + u_p(x) = -\frac{\cosh(\rho x)}{\alpha^2\cosh(\rho c)} + u_p(x) \end{aligned}$$

Remember for this setup $p_1(x) = u_p(x)$, so $p_1''(x) = \frac{2}{\alpha}$, so we get

$$v_f(x) = -\frac{p_1''(c)\cosh(x\sqrt{2\alpha})}{2\alpha\cosh(c\sqrt{2\alpha})} + p_1(x) = -\frac{\cosh(\rho x)}{\alpha^2\cosh(\rho c)} + u_p(x) = u(x).$$

Meaning the two value-functions agree with one another as $k_1 \to 0$.

3 N-players

Now suppose there are N policies, each controlling one object. From now on, let's refer to such a pair of policy and object as a "player". The goal for each player is to stay as close as possible to the other players.

This N-player problem can be formulated as follows. Let $\nu^i := \{(\tau_k^i, Y_k^i) : k \in \mathbb{N}\}$ be the policy for the ith player, where (as in the one-dimensional case), $\tau_1^i < \tau_2^i < \dots$ are stopping times denoting the impulse times for the ith player and Y_k^i is $\mathcal{F}_{\tau_k^i}$-measurable for each $k \in \mathbb{N}$ and indicates the kth impulse size for the ith player. Now, let $(X_t^1, \dots, X_t^N) \in \mathbb{R}^N$ be the positions of the players such that for $i = 1, \dots, N$,

$$\begin{aligned} X_t^i &= x^i + W_t^i + \xi_t^i, \\ \xi_t^i &= \sum_{k=1}^{\infty} I_{\{\tau_k^i \le t\}} Y_k^i, \end{aligned} \tag{3.1}$$

with $(X_{0-}^1, \dots, X_{0-}^N) = (x^1, \dots, x^N) =: \boldsymbol{x}$, where $(W_t^1, \dots, W_t^N)$ is an N-dimensional standard Brownian motion on $\mathbb{R}^N$ and ξ_t^i are the total aggregated impulses for the ith player up to time t.

Let $\overline{X}_t := \left(\sum_{i=1}^N X_t^i\right) \Big/ N$ be the moving average of our N-players. As in the one-dimensional case, the running cost function is $h(x) := x^2$ and $c(y,z) = k_1 + k_2|y - z|$. The goal for each player i is to minimize, over all admissible control policies $\boldsymbol{\nu} := (\nu^1, \dots, \nu^N) \in \mathcal{U}_N$, the following cost functional:

$$\begin{aligned} J^i(\boldsymbol{\nu}; \boldsymbol{x}) = \mathbb{E}\Bigg[&\int_0^\infty e^{-\alpha t} h\left(X_t^i - \overline{\boldsymbol{X}}_t\right) \mathrm{d}t \\ &+ \sum_{k=1}^{\infty} I_{\{\tau_k^i < \infty\}} e^{-\alpha \tau_k^i} c\left(X_{\tau_k^i -}^i, X_{\tau_k^i}^i\right) \Bigg]. \end{aligned} \tag{3.2}$$

The set $\mathcal{U}_N$ is defined as

$$\begin{aligned}\mathcal{U}_N := \Big\{ \big(\nu^1, \dots, \nu^N\big) \,\Big|\, & \nu^j \in \mathcal{U}_N^j, \mathbb{P}(\Delta\xi_t^i(\boldsymbol{x}) \cdot \Delta\xi_t^j(\boldsymbol{x})) = 0 \\ & \text{for any } t > 0, \boldsymbol{x} \in \mathbb{R}^N, i, j \in \{1, \dots, N\}, i \neq j \Big\},\end{aligned} \tag{3.3}$$

with

$$\mathcal{U}_N^j = \Big\{ \nu^j \,\Big|\, \exists\, \hat{\boldsymbol{\nu}} \text{ control policy s.t. } J^j((\hat{\boldsymbol{\nu}}^{-j}, \nu^j); \boldsymbol{x}) < \infty \Big\}. \tag{3.4}$$

Here $(\hat{\boldsymbol{\nu}}^{-j}, \nu^j) := (\hat{\nu}^1, \dots, \hat{\nu}^{j-1}, \nu^j, \hat{\nu}^{j+1}, \dots \hat{\nu}^N)$.

The condition in (3.3)

$$\mathbb{P}(\Delta\xi_t^i(\boldsymbol{x}) \cdot \Delta\xi_t^j(\boldsymbol{x})) = 0 \text{ for any } t > 0, \boldsymbol{x} \in \mathbb{R}^N, i, j \in \{1, \dots, N\}, i \neq j$$

means, we only consider policies where no two players act at the same time.

3.1 Nash equilibrium and HJB Equation

Definition 3.1 (Nash Equilibrium)**.** A tuple of admissible impulse controls $\boldsymbol{\nu}^* = (\nu^{1*}, \dots, \nu^{N*})$ is a Nash equilibrium (NE) of the N-player stochastic game if for any $i = 1, 2, \dots, N$, $\boldsymbol{X}_0 = \boldsymbol{x}$ and any $(\boldsymbol{\nu}^{-i*}, \nu^i)$, the following inequality holds:

$$J^i(\boldsymbol{\nu}^*; \boldsymbol{x}) \leq J^i((\boldsymbol{\nu}^{-i*}, \nu^i), \boldsymbol{x}).$$

$J^i(\boldsymbol{\nu}^*; \boldsymbol{x})$ is called the NE value associated with $\boldsymbol{\nu}^*$.

The first step to finding the NE solution is to derive and analyze the associated HJB system. To do that, let's define the action region $\mathcal{A}_i$ and waiting region $\mathcal{W}_i$ for each player.

Definition 3.2. Player i's *action region* $\mathcal{A}_i$ is defined as

$$\mathcal{A}_i := \{\boldsymbol{x} \in \mathbb{R}^N : \Delta\xi^i(\boldsymbol{x}) \neq 0\},$$

and her *waiting region* is defined as

$$\mathcal{W}_i := \mathbb{R}^N \setminus \mathcal{A}_i.$$

Denote $\mathcal{A}_{-i} := \bigcup_{j \neq i} \mathcal{A}_j$ the union of action regions of the other players and $\mathcal{W}_{-i} := \bigcap_{j \neq i} \mathcal{W}_j$ the common waiting region of the other players.

Introduce the operator $\mathcal{M}^i f(\boldsymbol{x}) := \inf_{y \in \mathbb{R}}\{f(\boldsymbol{x}^{-i}, y) + c(x_i, y)\}$. From the definition of $\mathcal{U}_N$ we know that $\mathcal{A}_i \cap \mathcal{A}_j = \emptyset$ for any $i \neq j$. With this the HJB equation for the NE for the N-player problem is

$$\begin{cases} \max\left\{\alpha w^i(\boldsymbol{x}) - h(x_i - \overline{\boldsymbol{x}}) - \mathcal{L}w^i(\boldsymbol{x}),\ w^i(\boldsymbol{x}) - \mathcal{M}^i w^i(\boldsymbol{x})\right\} = 0, & \forall \boldsymbol{x} \in \mathcal{W}_{-i}, \\ \dfrac{\partial}{\partial x_j} w^i(\boldsymbol{x}) = 0, & \forall \boldsymbol{x} \in \mathcal{A}_j, j \neq i, \end{cases} \tag{3.5}$$

where $\mathcal{L}w^i(\boldsymbol{x}) := \dfrac{1}{2}\displaystyle\sum_{j=1}^{N} \dfrac{\partial^2}{\partial x_j^2} w^i(\boldsymbol{x})$.

The derivation of (3.5) can be illustrated with the case of $N = 2$. In this case, if $\boldsymbol{x} = (x_1, x_2) \in \mathcal{A}_2$ and $\Delta\xi^{2*} \neq 0$. By the definition of the NE, player one is not expected to suffer a loss for otherwise she will have incentive to take action. Thus

$$w^1(x_1, x_2) = w^1(x_1, x_2 + \Delta\xi^{2*}).$$

Letting $\Delta\xi^{2*} \to 0$, we have $\frac{\partial}{\partial x_2} w^1(\boldsymbol{x}) = 0$ for $\boldsymbol{x} \in \mathcal{A}_2$.

If $\boldsymbol{x} \in \mathcal{W}_2$, $\Delta\xi^{2*}(\boldsymbol{x}) = 0$, then the control problem for player one becomes a classical single player impulse control problem. Therefore $w^1(\boldsymbol{x})$ satisfies

$$\max\left\{\alpha w^1(\boldsymbol{x}) - h\left(\frac{x_1 - x_2}{2}\right) - \frac{1}{2}\sum_{j=1}^{2}\frac{\partial^2}{\partial x_j^2}w^1(\boldsymbol{x}),\ w^1(\boldsymbol{x}) - \mathcal{M}^1 w^1(\boldsymbol{x})\right\} = 0, \quad \forall \boldsymbol{x} \in \mathcal{W}_2.$$

3.2 Verification Theorem

Theorem 3.3 (Verification Theorem)**.** *Suppose $\boldsymbol{\nu}^* = (\nu^{1*}, \ldots, \nu^{N*})$ is an admissible policy and the corresponding value $w^i(\boldsymbol{x}) := J^i(\boldsymbol{\nu}^*; \boldsymbol{x})$ satisfies*

(i) *the function $w^i(\boldsymbol{x}) \in C^2(\overline{\mathcal{W}_{-i}})$ and satisfies* (3.5)*,*

(ii) *for any admissible policy ν^i, the controlled dynamic $(\boldsymbol{X}_t^{-i*}, X_t^i)$ under the policy $(\boldsymbol{\nu}^{-i*}, \nu^i)$ stays in $\mathcal{W}_{-i}$ for all $t \geq 0$ $\mathbb{P}$-a.s.,*

(iii) *there exists a function $u^i(\boldsymbol{x}) \in C^2(\mathbb{R}^N)$ such that $u^i(\boldsymbol{x}) = w^i(\boldsymbol{x})$ on $\overline{\mathcal{W}-i}$,*

(iv) *there exists an increasing sequence of stopping times $\{\beta_n : n \in \mathbb{N}\}$ with $\beta_n \to \infty$ a.s. such that*

$$M_{t\wedge\beta_n} := \int_0^{t\wedge\beta_n} e^{-\alpha s}\sum_{j=1}^{N}\frac{\partial}{\partial x_j}u^i(\boldsymbol{X}_s^{-i*}, X_s^i)\mathrm{d}W_s^j$$

is a martingale with mean zero; moreover the transversality condition holds:

$$\lim_{T\to\infty}\lim_{n\to\infty}\mathbb{E}\left[\mathrm{e}^{-\alpha(T\wedge\beta_n)}u^i(\boldsymbol{X}_{T\wedge\beta_n}^{-i*}, X_{T\wedge\beta_n}^i)\right] = 0,$$

Then $\boldsymbol{\nu}^*$ is a NE with value u^i.

Proof. Under the policy $(\boldsymbol{\nu}^{-i*}, \nu^i)$, we have

$$X_t^i = x^i + W_t^i + \xi_t^i, \quad t \geq 0,$$

and for $j \neq i$

$$X_t^{j*} = x^j + W_t^j + \xi_t^{j*}, \quad t \geq 0.$$

Let β_n as in the statement of the theorem. We now apply Itô's formula to the process $e^{-\alpha t} u^i(\boldsymbol{X}_t^{-i*}, X_t^i)$ to obtain

$$\begin{aligned}
&\mathbb{E}\Big[e^{-\alpha T\wedge\beta_n} u^i\left(\boldsymbol{X}_{T\wedge\beta_n}^{-i*}, X_{T\wedge\beta_n}^i\right)\Big] - u^i(\boldsymbol{x}) \qquad (3.6)\\
&= \mathbb{E}\Bigg[\int_0^{T\wedge\beta_n} e^{-\alpha t}\bigg(\mathcal{L}u^i\left(\boldsymbol{X}_t^{-i*}, X_t^i\right) - \alpha u^i\left(\boldsymbol{X}_t^{-i*}, X_t^i\right)\bigg)\mathrm{d}t\Bigg]\\
&+ \mathbb{E}\Bigg[\sum_{k=1}^{\infty} I_{\{\tau_k^i \leq T\wedge\beta_n\}} e^{-\alpha\tau_k^i}\Big[u^i\left(\boldsymbol{X}_{\tau_k^i}^{-i*}, X_{\tau_k^i}^i\right) - u^i\left(\boldsymbol{X}_{\tau_k^i-}^{-i*}, X_{\tau_k^i-}^i\right)\Big]\Bigg]\\
&+ \mathbb{E}\Bigg[\sum_{j=1, j\neq i}^{N}\sum_{k=1}^{\infty} I_{\{\tau_k^{j*} \leq T\wedge\beta_n\}} e^{-\alpha\tau_k^{j*}}\Big[u^i\left(\boldsymbol{X}_{\tau_k^{j*}}^{-i*}, X_{\tau_k^{j*}}^i\right) - u^i\left(\boldsymbol{X}_{\tau_k^{j*}-}^{-i*}, X_{\tau_k^{j*}-}^i\right)\Big]\Bigg].
\end{aligned}$$

By condition (ii), the last term is equal to zero. On the other hand, by conditions (ii) and (i), we have

$$\mathcal{L}u^i\left(\boldsymbol{X}_t^{-i*}, X_t^i\right) - \alpha u^i\left(\boldsymbol{X}_t^{-i*}, X_t^i\right) \geq -h\left(X_t^i - \overline{\boldsymbol{X}}_t\right).$$

Since $\mathcal{A}_i \cap \mathcal{A}_j = \emptyset$ for $i \neq j$, we have $\boldsymbol{X}_{\tau_k^i}^{-i*} = \boldsymbol{X}_{\tau_k^i-}^{-i*}$. This, together with conditions (ii) and (i), implies that

$$u^i\left(\boldsymbol{X}_{\tau_k^i}^{-i*}, X_{\tau_k^i}^i\right) - u^i\left(\boldsymbol{X}_{\tau_k^i-}^{-i*}, X_{\tau_k^i-}^i\right) \geq -c\left(X_{\tau_k^i-}^i, X_{\tau_k^i}^i\right).$$

Plugging the above two inequalities into (3.6), we obtain

$$\begin{aligned}
&\mathbb{E}\Bigg[e^{-\alpha T\wedge\beta_n}u^i\left(\boldsymbol{X}^{-i*}_{T\wedge\beta_n}, X^i_{T\wedge\beta_n}\right)\Bigg] - u^i(\boldsymbol{x}) \\
&\quad \geq -\mathbb{E}\Bigg[\int_0^{T\wedge\beta_n} e^{-\alpha t}h\left(X^i_t - \overline{\boldsymbol{X}}_t\right)\mathrm{d}t\Bigg] \\
&\qquad - \mathbb{E}\Bigg[\sum_{k=1}^{\infty} I_{\{\tau^i_k \leq T\wedge\beta_n\}}e^{-\alpha\tau^i_k}c\left(X^i_{\tau^i_k-}, X^i_{\tau^i_k}\right)\Bigg].
\end{aligned}$$

That is

$$\begin{aligned}
u^i(\boldsymbol{x}) \leq \mathbb{E}&\Bigg[e^{-\alpha T\wedge\beta_n}u^i\left(\boldsymbol{X}^{-i*}_{T\wedge\beta_n}, X^i_{T\wedge\beta_n}\right) + \sum_{k=1}^{\infty} I_{\{\tau^i_k \leq T\wedge\beta_n\}}e^{-\alpha\tau^i_k}c\left(X^i_{\tau^i_k-}, X^i_{\tau^i_k}\right)\Bigg] \\
&+ \mathbb{E}\Bigg[\int_0^{T\wedge\beta_n} e^{-\alpha t}h\left(X^i_t - \overline{\boldsymbol{X}}_t\right)\mathrm{d}t\Bigg].
\end{aligned}$$

Since $h, c \geq 0$, we can use the monotone convergence theorem and the transversality condition in (iii) to obtain

$$\begin{aligned}
u^i(\boldsymbol{x}) &\leq \mathbb{E}\Bigg[\sum_{k=1}^{\infty} I_{\{\tau^i_k < \infty\}}e^{-\alpha\tau^i_k}c\left(X^i_{\tau^i_k-}, X^i_{\tau^i_k}\right) \\
&\qquad\quad + \int_0^{\infty} e^{-\alpha t}h\left(X^i_t - \overline{\boldsymbol{X}}_t\right)\mathrm{d}t\Bigg] \\
&= J^i\left((\boldsymbol{\nu}^{-i*}, \nu^i); \boldsymbol{x}\right).
\end{aligned}$$

This finishes the proof. □

3.3 Finding the Optimal Policies

Now that we have the Verification Theorem, we can start finding the policies for our players. Let's try a similar approach to the one-dimensional case, meaning our player idles as long as she stays within a certain distance of the moving average and jumps closer to it, if she is farther away.

Note 3.4. *The distance of a player to the moving average can also be expressed in terms of the distance from the player to the average of the other players (excluding the player itself):*

$$X_t^i - \overline{\boldsymbol{X}}_t = X_t^i - \frac{\sum_{j=1}^N X_t^j}{N} = \frac{N-1}{N}\left(X_t^i - \frac{\sum_{j\neq i} X_t^j}{N-1}\right) = \frac{N-1}{N}\left(X_t^i - \overline{\boldsymbol{X}_t^{-i}}\right) \tag{3.7}$$

Using this, we can also formulate that threshold in terms of the distance to the other players' average. So we can say there exists some constant $y_{N*} > 0$, s.t. we can decompose the ith players action and waiting region as follows:

$$\mathcal{A}_i := \left\{E_i^- \cup E_i^+\right\} \cap Q_i, \qquad \mathcal{W}_i = \mathbb{R}^N \setminus \mathcal{A}_i, \tag{3.8}$$

where

$$\begin{aligned} E_i^- &:= \left\{\boldsymbol{x} \in \mathbb{R}^N : d_i^N(\boldsymbol{x}) < -y_{N*}\right\}, \\ E_i^+ &:= \left\{\boldsymbol{x} \in \mathbb{R}^N : d_i^N(\boldsymbol{x}) > y_{N*}\right\}, \\ d_i^N(\boldsymbol{x}) &:= x^i - \overline{\boldsymbol{x}^{-i}}. \end{aligned} \tag{3.9}$$

Q_i is a partition of $\mathbb{R}^N$. We need that, since we want to avoid any two players sharing an action region, which could happen if we just used $A_i = E_i^- \cup E_i^+$. This is especially noticeable in the case $N = 2$, because of the symmetry we get there:

$$d_1^2(x^1, x^2) = \frac{1}{2}(x^1 - x^2) = -\frac{1}{2}(x^2 - x^1) = -d_2^2(x^1, x^2)$$

So $E_1^- = E_2^+$ and $E_1^+ = E_2^-$. In general we can use the partition

$$Q_i := \left\{ \boldsymbol{x} \in \mathbb{R}^N \middle| \quad \left|d_i^N(\boldsymbol{x})\right| \geq \left|d_k^N(\boldsymbol{x})\right| \text{ for any } k < i; \quad \left|d_i^N(\boldsymbol{x})\right| > \left|d_k^N(\boldsymbol{x})\right| \text{ for any } k > i \right\}, \tag{3.10}$$

i.e. we use the rule: Should more than one player be in their "action region", the one furthest from the center pushes first. If multiple players have the biggest distance, the one with the highest index pushes first.

On the common waiting region of all players $\mathcal{W} = \bigcap_{i=1}^N \mathcal{W}_i$, every policy idles, so every candidate solution w^i should satisfy:

$$\alpha w^i(\boldsymbol{x}) - h(\boldsymbol{x}) - \mathcal{L}w^i(\boldsymbol{x}) = 0$$

To solve this, we start by finding the homogeneous solution w_h^i to $\alpha w_h^i(\boldsymbol{x}) - \mathcal{L}w_h^i(\boldsymbol{x}) = 0$, by assuming $w_h^i(\boldsymbol{x}) = e^{\lambda \cdot d_N^i(\boldsymbol{x})}$:

$$\begin{aligned} \alpha - \frac{1}{2}\left[\lambda^2 + (N-1)\left(\frac{\lambda}{N-1}\right)^2\right] &= 0 \\ \alpha - \frac{1}{2}\lambda^2\left[1 + \frac{1}{N-1}\right] &= 0 \\ \lambda = \pm\sqrt{2\alpha \cdot \frac{(N-1)}{N}} &=: \pm\rho_N \end{aligned}$$

So our homogeneous solution is given by

$$w_h^i(\boldsymbol{x}) = B_N^i e^{\rho_N \cdot d_N^i(\boldsymbol{x})} + C_N^i e^{-\rho_N \cdot d_N^i(\boldsymbol{x})},$$

for some constants $B_N^i, C_N^i \in \mathbb{R}$.

We find the particular solution w_p^i by assuming $w_p^i(\boldsymbol{x}) = ad_N^i(\boldsymbol{x})^2 + bd_N^i(\boldsymbol{x}) + c.$

$$\begin{aligned} \alpha w_p^i(\boldsymbol{x}) &= h(x^i - \overline{\boldsymbol{x}}) + \mathcal{L} w_p^i(\boldsymbol{x}) \\ \alpha \left(ad_N^i(\boldsymbol{x})^2 + bd^i(\boldsymbol{x}) + c \right) &= \left(\frac{N-1}{N} d_N^i(\boldsymbol{x}) \right)^2 + \frac{1}{2} \left(2a + \frac{2a}{(N-1)^2}(N-1) \right) \\ \alpha \left(ad_N^i(\boldsymbol{x})^2 + bd_N^i(\boldsymbol{x}) + c \right) &= \left(\frac{N-1}{N} \right)^2 d_N^i(\boldsymbol{x})^2 + a \left(1 + \frac{1}{(N-1)} \right) \\ a = \frac{(N-1)^2}{\alpha N^2} \qquad b &= 0 \qquad c = \frac{a}{\alpha} \cdot \frac{N}{(N-1)} = \frac{N-1}{\alpha^2 N} \end{aligned}$$

Let $\alpha_N := \frac{\alpha N^2}{(N-1)^2}$, so $a = \frac{1}{\alpha_N}$ and $c = \frac{N^3}{\alpha_N^2 (N-1)^3}$. Then we have

$$w^i(\boldsymbol{x}) = w_h^i(\boldsymbol{x}) + w_p^i(\boldsymbol{x}) = B_N^i e^{\rho_N \cdot d_N^i(\boldsymbol{x})} + C_N^i e^{-\rho_N \cdot d_N^i(\boldsymbol{x})} + \frac{d_N^i(\boldsymbol{x})^2}{\alpha_N} + \frac{N^3}{\alpha_N^2 (N-1)^3} \tag{3.11}$$

for any $\boldsymbol{x} \in \mathcal{W}$.

Let's take a closer look at the case $N = 2$. Because of the above mentioned symmetry, we get the following result if we use the partition from (3.10):

$$\begin{aligned} Q_1 &= \left\{ \boldsymbol{x} \in \mathbb{R}^2 \middle| \quad \left| d_1^2(\boldsymbol{x}) \right| > \left| d_2^2(\boldsymbol{x}) \right| \right\} = \emptyset \\ Q_2 &= \left\{ \boldsymbol{x} \in \mathbb{R}^2 \middle| \quad \left| d_2^2(\boldsymbol{x}) \right| \geq \left| d_1^2(\boldsymbol{x}) \right| \right\} = \mathbb{R}^2. \end{aligned}$$

In other words, player 2 does all the work, while player one always idles. This also gives us

$$\begin{aligned} &\mathcal{W}_1 = \mathbb{R}^2, && \mathcal{A}_1 = \emptyset, \\ &\mathcal{W}_2 = \left\{ (x^1, x^2) \in \mathbb{R}^2 : |x^2 - x^1| \leq y_{2*} \right\}, && \mathcal{A}_2 = E_2^- \cup E_2^+ \\ &E_2^- = \left\{ (x^1, x^2) \in \mathbb{R}^2 : x^2 - x^1 < -y_{2*} \right\}, && E_2^+ = \left\{ (x^1, x^2) \in \mathbb{R}^2 : x^2 - x^1 > y_{2*} \right\} \end{aligned}$$

To find the NE for this partition, let's start with finding player two's candidate function w^2. Notice that, since player two is doing all the work, player one's action region is the empty set, so the second condition in (3.5), i.e. $\dfrac{\partial}{\partial x^1} w^2(x^1, x^2) = 0 \quad \forall (x^1, x^2) \in \mathcal{A}_1$, is actually no constraint at all. This means we only have to consider the first part, making this problem very similar to the one-dimensional case.

From (3.11), we already know how w^2 has to look like on $\mathcal{W}_2 = \mathcal{W}$. Furthermore, from the symmetry of this partition, we get $w^2(\boldsymbol{x}) = w^2(\boldsymbol{y})$ if $|d_2^2(\boldsymbol{x})| = |d_2^2(\boldsymbol{y})|$. This also gives us that $A_2^2 = B_2^2$, so with $A_2^2 := B_2^2/2$, we get

$$w^2(x^1, x^2) = A_2^2 \cosh\big(\rho_2 (x^2 - x^1)\big) + \frac{(x^2 - x^1)^2}{\alpha_2} + \frac{8}{\alpha_2^2} \qquad \forall (x^1, x^2) \in \mathcal{W}_2. \tag{3.12}$$

On $\mathcal{A}_2$, we know that player two pushes, so

$$\begin{aligned} w^2(x^1, x^2) &= w^2(x^1, x^1 - z_{2*}) + c(x^2, x^1 - z_{2*}) \qquad \forall (x^1, x^2) \in E_2^-, \\ w^2(x^1, x^2) &= w^2(x^1, x^1 + z_{2*}) + c(x^2, x^1 + z_{2*}) \qquad \forall (x^1, x^2) \in E_2^+, \end{aligned}$$

for some constant z_{2*}. As in the one dimensional case, z_{2*} should be the optimal jump distance, so

$$\begin{aligned} \frac{\partial}{\partial z} \left[w^2\left(x^1, x^1 + z_{2*}\right) + c\left(x^1, x^2 + z_{2*}\right) \right] &= 0 \\ \rho_2 A_2^2 \sinh(\rho_2 z_{2*}) + \frac{2}{\alpha_2} z_{2*} - k_2 &= 0. \end{aligned}$$

Furthermore, using the smooth-pasting technique along $x^2 = x^1 + y_{2*}$, we get

$$\begin{aligned} w^2(x^1, x^1 + y_{2*}-) &= A_2^2 \cosh\left(\rho_2 y_{2*}\right) + \frac{y_{2*}^2}{\alpha_2} + \frac{8}{\alpha_2^2} \\ &= A_2^2 \cosh\left(\rho_2 z_{2*}\right) + \frac{z_{2*}^2}{\alpha_2} + \frac{8}{\alpha_2^2} + c\left(y_{2*}, z_{2*}\right) = w^2(x^1, x^1 + y_{2*}+) \\ \frac{\partial}{\partial x^2} w^i(x^1, x^1 + y_{2*}-) &= \rho_2 A_2^2 \sinh(\rho_2 y_{2*}) + \frac{2}{\alpha_2} y_{2*} = k_2 = \frac{\partial}{\partial x^2} w^2(x^1, x^1 + y_{2*}+) \end{aligned}$$

Giving us the following system of equalities:

$$A_2^2 = \frac{c(y_{2*}, z_{2*}) + \frac{z_{2*}^2 - y_{2*}^2}{\alpha_2}}{\cosh(\rho_2 y_{2*}) - \cosh(\rho_2 z_{2*})} = \frac{\alpha_2 k_2 - 2y_{2*}}{\alpha_2 \rho_2 \sinh(\rho_2 y_{2*})} = \frac{\alpha_2 k_2 - 2z_{2*}}{\alpha_2 \rho_2 \sinh(\rho_2 z_{2*})}.$$

If we compare this to the system we found for the one-dimensional case

$$A = \frac{c(y_*, z_*) + \frac{z_*^2 - y_*^2}{\alpha}}{\cosh(\rho y_*) - \cosh(\rho z_*)} = \frac{k_2 \alpha - 2y_*}{\alpha \cdot \rho \sinh(\rho y_*)} = \frac{k_2 \alpha - 2z_*}{\alpha \cdot \rho \sinh(\rho z_*)},$$

we can see that they are very similar and can be solved the same way. Unfortunately, as with the one dimensional case, there is no simple analytic expression for A_2^2, z_{2*} and y_{2*}, but finding the solution numerically works nearly the same way as for the one dimensional case (See Appendix Python-Code N-D). All put together we get

$$w^2(x^1, x^2) = \begin{cases} w^2(x^1, x^1 - z_{2*}) + c(x^2, x^1 - z_{2*}) & (x^1, x^2) \in E_2^-, \\ A_2^2 \cosh\left(\rho_2 \cdot (x^2 - x^1)\right) + \frac{(x^2 - x^1)^2}{\alpha_2} + \frac{8}{\alpha_2^2} & (x^1, x^2) \in \mathcal{W}_2, \\ w^2(x^1, x^1 + z_{2*}) + c(x^2, x^1 + z_{2*}) & (x^1, x^2) \in E_2^+. \end{cases} \tag{3.13}$$

With $u^2(x^1, x^2) = A_2^2 \cosh\left(\sqrt{\alpha} \cdot (x^2 - x^1)\right) + \frac{(x^2 - x^1)^2}{4\alpha} + \frac{1}{2\alpha^2}$, we can check as in the one dimensional case that w^2 fulfills the conditions in the Verification Theorem, meaning to find a NE solution, all that's left is to find w^1.

As with w^2, we know w^1 on $\mathcal{W}$ is given by (3.11) and with $A_2^1 := B_2^1/2$ and the symmetry we get

$$w^1(x^1,x^2) = A_2^1 \cosh\big(\rho_2(x^1-x^2)\big) + \frac{(x^1-x^2)^2}{\alpha_2} + \frac{8}{\alpha_2^2} \qquad \forall (x^1,x^2) \in \mathcal{W}. \tag{3.14}$$

On $\mathcal{A}_2$ we know that player two pushes and from (3.5) we know that player one incurs no cost from that push. So we should expect

$$\begin{aligned} w^1(x^1,x^2) &= w^1(x^1,x^1-z_{2*}) \qquad \forall (x^1,x^2) \in \mathbb{E}_2^-, \\ w^1(x^1,x^2) &= w^1(x^1,x^1+z_{2*}) \qquad \forall (x^1,x^2) \in \mathbb{E}_2^+. \end{aligned}$$

Matching the values along $x^2 = x^1 + y_{2*}$, we get

$$\begin{aligned} w^1(x^1,x^1+y_{2*}-) &= A_2^1 \cosh\left(\rho_2 y_{2_*}\right) + \frac{y_{2*}^2}{\alpha_2} + \frac{8}{\alpha_2^2} \\ &= A_2^1 \cosh\left(\rho_2 z_{2*}\right) + \frac{z_{2*}^2}{\alpha_2} + \frac{8}{\alpha_2^2} = w^1(x^1,x^1+y_{2*}+) \end{aligned}$$

Solving for A_2^1 gives us the last missing variable:

$$A_2^1 = \frac{\left(z_{2*}^2 - y_{2*}^2\right)/\alpha_2}{\cosh(\rho_2 y_{2*}) - \cosh(\rho_2 z_{2*})}.$$

So the final expression for w^1 is

$$w^1(x^1,x^2) = \begin{cases} w^1(x^1,x^1-z_{2*}) & (x^1,x^2) \in E_2^-, \\ A_2^1 \cosh\left(\rho_2 \cdot (x^1-x^2)\right) + \frac{(x^1-x^2)^2}{\alpha_2} + \frac{8}{\alpha_2^2} & (x^1,x^2) \in \mathcal{W}_2, \\ w^1(x^1,x^1+z_{2*}) & (x^1,x^2) \in E_2^+. \end{cases} \tag{3.15}$$

Let's check if w^1 satisfies the HJB equation (3.5).

Per construction only $w^1(x^1,x^2) - \mathcal{M}^1 w^1(x^1,x^2) \leq 0$ on $\mathcal{W}$ is left to check. Let $q \in \mathbb{R}$, then we need to check whether

$$w^1(x^1,x^2) - w^1(x^1+q,x^2) - c(x^1,x^1+q) \leq 0.$$

Suppose $(x^1+q,x^2) \in E_2^+$, then $w^1(x^1+q,x^2) = w^1(x^1+q,x^1+q+z_{2*})$. From the definition of A_2^1 and the symmetry of w^1 around $x^1 = x^2$, we get

$$w^1(x^1+q,x^1+q+z_{2*}) = w^1(x^1+q,x^1+q+y_{2*}) = w^1(x^2-y_{2*},x^2).$$

But since $(x^1+q,x^2) \in E_2^+$, we also know that $x^1+q < x^2 - y_{2*} \leq x^1$, so

$$w^1(x^1,x^2) - w^1(x^1+q,x^2) - c(x^1,x^1+q) < w^1(x^1,x^2) - w^1(x^2-y_{2*},x^2) - c(x^1,x^2-y_{2*}).$$

Using the same argument for $(x^1,x^2) \in E_2^-$, we see that it is enough to check any q, s.t. $(x^1+q,x^2) \in \mathcal{W}$. Then

$$\begin{aligned} w^1(x^1,x^2) - & w^1(x^1+q,x^2) - c(x^1,x^1+q) \\ &= \left[A_2^1 \cosh\left(\sqrt{\alpha}\cdot(x^1-x^2)\right) + \frac{(x^1-x^2)^2}{4\alpha} + \frac{1}{2\alpha^2} \right] \\ &- \left[A_2^1 \cosh\left(\sqrt{\alpha}\cdot(x^1+q-x^2)\right) + \frac{(x^1+q-x^2)^2}{4\alpha} + \frac{1}{2\alpha^2} + k_1 + k_2|q| \right] \overset{?}{\leq} 0 \end{aligned}$$

Let $y = x^1 - x^2$ and $z = x^1 + c - x^2$. Then, if $|z| < |y|$ we get

$$A_2^1 \overset{?}{\leq} \frac{k_1 + k_2|z-y| + \frac{z^2-y^2}{\alpha_2}}{\cosh(\rho_2 y) - \cosh(\rho_2 z)}$$

This is true, since $A_2^1 < A_2^2$ and A_2^2 is the minimum of the right expression (as in the one dimensional case).

Now suppose $|z| > |y|$. This case is more tricky to show and we actually have no analytic proof yet, which will have to be done in a future work. For now, we verified that the inequality holds numerically and will assume its correctness from here on.

From here it is straightforward to check that w^1 satisfies the Verification Theorem, so we have found a possible NE Solution. Let

$$\begin{aligned}
&\nu^{1*} := \text{"Zero-control"}\ , \qquad \nu^{2*} := (\tau_k^{2*}, Y_k^{2*}), \\
&\tau_0^{2*} := \inf\left\{t > 0 : |X_t^2 - X_t^1| > y_{2*}\right\} \\
&\tau_k^{2*} := \inf\left\{t > \tau_{k-1}^{2*} : |X_t^2 - X_t^1| > y_*\right\} \qquad\qquad \forall k \in \mathbb{N}_> \\
&Y_k^{2*} := sgn\left((X^2 - X^1)_{\tau_n^*-}\right) \cdot z_* - (X^2 - X^1)_{\tau_n^*-} \quad \forall k \in \mathbb{N}_\geq
\end{aligned} \tag{3.16}$$

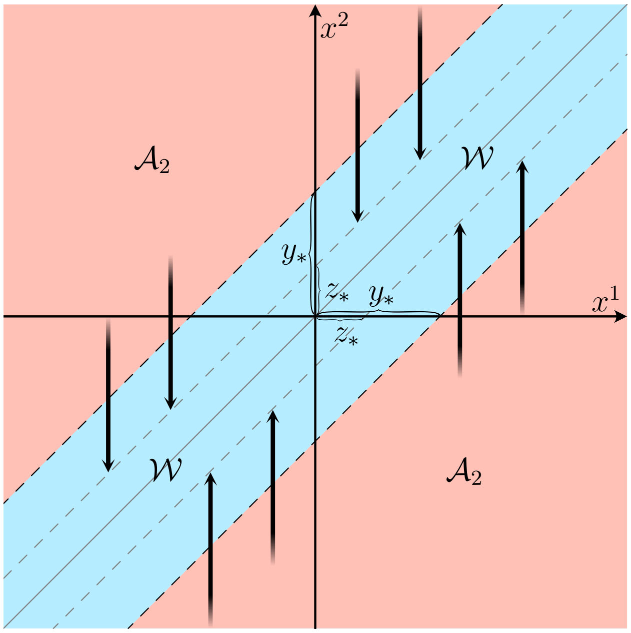


Figure 7: Possible NE impulse control strategy for 2D

Theorem 3.5. *Let $\boldsymbol{\nu}^* = (\nu^{1*}, \nu^{2*})$ defined as in (3.16), then $\boldsymbol{\nu}$ is a NE with values w^1 and w^2 defined as in* (3.13) *and* (3.15) *respectively.*

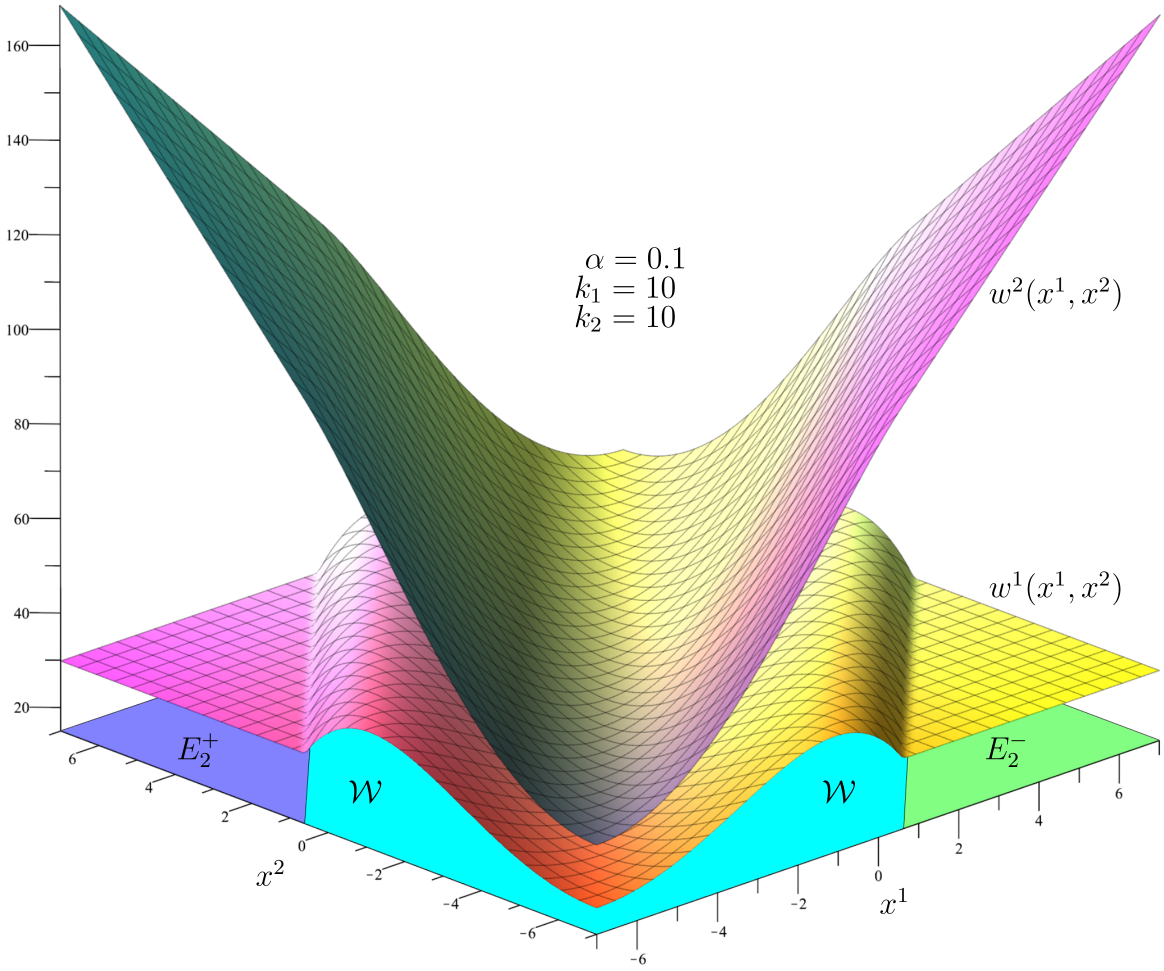


Figure 8: Sample plot of w^1 and w^2 - NE solutions

In the above figure, we can see a sample plot of w^1 and w^2. The figure in the next page shows a cross-section of the above plot for $(x^1 = -x^2)$ that shows what influence z_{2*} and y_{2*} have on the functions.

As we can see in both plots, $w^1 < w^2$ everywhere. This might seem unexpected at first, but does make sense. Player one never has to push, so the total cost should be lower than for player two for the same start position.

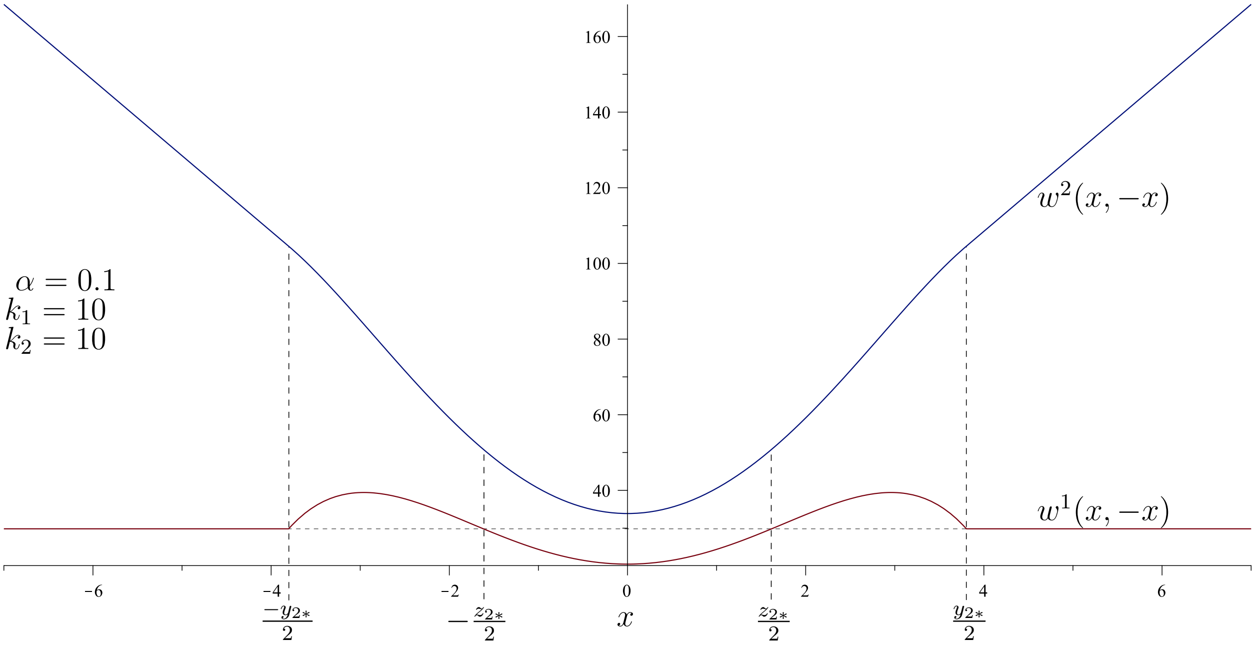


Figure 9: Cross section of w^1 and w^2 - NE solutions

Because of the symmetry in the two dimensional case, we actually have more than one NE solution. One that we can immediately find, is if we flip the roles of player one and player two, so now player one always pushes and player two always idles. This solution also corresponds to the partition $Q_1 = \mathbb{R}^2, \qquad Q_2 = \emptyset$.

3.4 Outlook

Another partition that might be interesting to analyze in a future work is

$$Q_1 = \left\{ (x^1, x^2) \in \mathbb{R}^2 \middle| \quad x^1 > x^2 \right\}$$
$$Q_2 = \left\{ (x^1, x^2) \in \mathbb{R}^2 \middle| \quad x^1 \leq x^2 \right\}$$

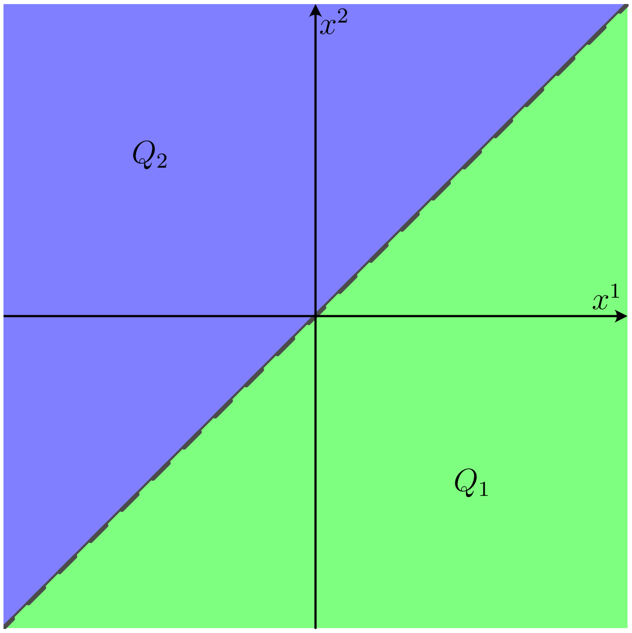


Figure 10: Partition of $\mathbb{R}^2$

This partition would result in both player one and player two doing pushes, with player one to the right of $x^1 = x^2 - y_{2*}$ and player one to the left of $x^2 = x^1 - y_{2*}$. This also means that we would loose the inherent symmetry the problem had until now, meaning the constant B_2^i and C_3^i in (3.11) would most likely not be the same anymore. A possible cross-section of the corresponding w^1 and w^2 functions could look as in Figure 11.

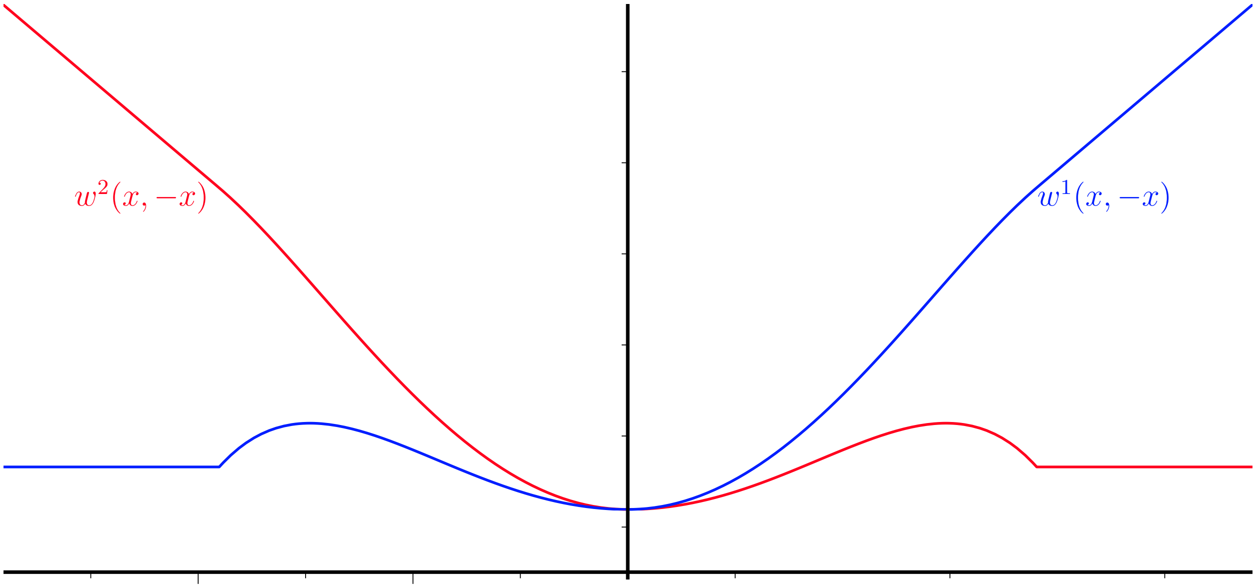


Figure 11: Possible shape of cross section of w^1 and w^2 for non-symmetric strategy

After finishing with the two dimensional case, the next step is of course to get a general NE solution for the N-player case. One thing that could cause problems there, compared to before is that $\overline{\mathcal{A}_i} \cap \overline{\mathcal{A}_j} \neq \emptyset$ in general, meaning the action-regions share borders. Matching the different functions along those borders could prove difficult. Another thing to consider, is that pushes from one player may cause another player to react. Consider the following 5-player situation. $\boldsymbol{X}_{0-} = (-\frac{2}{3}y_{5*} - 0.1, -\frac{2}{3}y_{5*}, \frac{2}{3}y_{5*}, \frac{2}{3}y_{5*}, \frac{2}{3}y_{5*})$, then the distance of player one to the others is:

$$d_5^1(\boldsymbol{X}_{0-}) = -\frac{2}{3}y_{5*} - 0.1 - \frac{-\frac{2}{3}y_{5*} + \frac{2}{3}y_{5*} + \frac{2}{3}y_{5*} + \frac{2}{3}y_{5*}}{4} = -y_{5*} - 0.1 < -y_{5*}.$$

So player one has incentive to act and makes a push toward the center. But this causes player two to slip out of the waiting region, resulting in a push from player two. This goes against the assumption that actions of other players should not result in a disadvantage and definitely warrants a closer look in a future work.

4 References

[1] KL Helmes, RH Stockbridge and C Zhu, *Impulse Control of Standard Brownian Motion: Discounted Criterion*, System Modeling and Optimization, C Pötzsche, C Heuberger, B Kaltenberger and F Rendl (eds.), (2014), 158–169, Springer.

[2] X Guo and R Xu, *Stochastic games for fuel follower problem: N versus mean field game*, SIAM Journalon Control and Optimization, (2019), 57(1):659–692.

Appendix Python-Code 1-D

```python
import numpy as np
import scipy.optimize as opt
import matplotlib.pyplot as plt

# set as needed
k1 = 10       # > 0
k2 = 10       # > 0
alpha = 0.1   # > 0

rho = np.sqrt(2*alpha)

def A(z, y):
    try:
        return (k1 + k2*np.abs(y-z) + (z**2 - y**2)/alpha)/(np.cosh(rho*y) - np.cosh(rho*z))
    except OverflowError:
        return np.inf

def f(x):
    return (2*x - k2*alpha)/(np.sinh(rho*x))

hat_x = opt.minimize_scalar(lambda x: -f(x)).x
hat_t = f(hat_x)

def inv_f(t, left_of_hat_x=True):
    # method: find zero for f(x) - t
    if left_of_hat_x:
        # we know it has to cross zero between 0 and hat_x
        bounds = [0, hat_x]
    else:
        # find the zero crossing
        b1, b2 = 0, 10
        while f(hat_x + b2) >= t:
            b1 = b2
            b2 *= 2
        bounds = [hat_x + b1, hat_x + b2]
    return opt.brentq(lambda x: f(x) - t, bounds[0], bounds[1])

def P(t):
    p_z = inv_f(t, left_of_hat_x=True)
    p_y = inv_f(t, left_of_hat_x=False)

    return (p_z, p_y)

def hat_A(t):
    z, y = P(t)
    return A(z, y)

#print result
t_star = opt.minimize_scalar(hat_A, bounds=(0, hat_t), method='bounded').x
z_star, y_star = P(t_star)
print(f'A = {hat_A(t_star)}; z* = {z_star}; y*={y_star}')

#plot result
ts = np.arange(0, 1, 0.005) * hat_t
res = [hat_A(t) for t in ts]
plt.plot(ts, res, '-', [t_star], [A(z_star, y_star)], 'ro')
plt.show()
```

Appendix Python-Code N-D

```python
import numpy as np
import scipy.optimize as opt
import matplotlib.pyplot as plt

# set as needed
k1 = 10       # > 0
k2 = 10       # > 0
alpha = 0.1   # > 0
N = 2         # > 1

rhoN   = np.sqrt(2*alpha * (N-1)/N)
alphaN = (alpha * N**2)/(N-1)**2

def A(z, y):
    try:
        return (k1 + k2*np.abs(y-z) + (z**2 - y**2)/alphaN)/(np.cosh(rhoN*y) - np.cosh(rhoN*z))
    except OverflowError:
        return np.inf

def f(x):
    return (2*x - k2*alphaN)/(np.sinh(rhoN*x))

hat_x = opt.minimize_scalar(lambda x: -f(x)).x
hat_t = f(hat_x)

def inv_f(t, left_of_hat_x=True):
    # method: find zero for f(x) - t
    if left_of_hat_x:
        # we know it has to cross zero between 0 and hat_x
        bounds = [0, hat_x]
    else:
        # find the zero crossing
        b1, b2 = 0, 10
        while f(hat_x + b2) >= t:
            b1 = b2
            b2 *= 2
        bounds = [hat_x + b1, hat_x + b2]
    return opt.brentq(lambda x: f(x) - t, bounds[0], bounds[1])

def P(t):
    p_z = inv_f(t, left_of_hat_x=True)
    p_y = inv_f(t, left_of_hat_x=False)

    return (p_z, p_y)

def hat_A(t):
    z, y = P(t)
    return A(z, y)

#print result
t_star = opt.minimize_scalar(hat_A, bounds=(0, hat_t), method='bounded').x
z_star, y_star = P(t_star)
print(f'A = {hat_A(t_star)}; z* = {z_star}; y*={y_star}')

#plot result
ts = np.arange(0, 1, 0.005) * hat_t
res = [hat_A(t) for t in ts]
plt.plot(ts, res, '-', [t_star], [A(z_star, y_star)], 'ro')
plt.show()
```