

Decentralized Control for Coordinated flow of Multi-Agent Systems

Valentino Crespi, George Cybenko, Daniela Rus, Massimo Santini
Institute for Security Technology Studies
Dartmouth College, Hanover, NH 03755 USA
{valentino.crespi,gvc, rus, santini}@dartmouth.edu
Dartmouth Computer Science Technical Report TR2002-414

Abstract

This paper describes a distributed algorithm for coordinating the flow of a mass of vehicles approaching a highway exit or a tollbooth. We provide the problem formulation, a general methodology for distributed control and an instantiation of this methodology to the coordinated flow problem. We analyze our algorithm and provide experimental data.

1 Introduction

We are interested in issues related to the *navigation* and *coordination* control in multi-agent systems¹. In this paper we discuss a class of problems that admit a decentralized control solution designed starting from a classical centralized solution and replacing global resources with local resources. We believe that under particular conditions the performance analysis of the centralized solution is preserved under such a transformation.

We consider a class of problems we call the “Opera Problem” [1]. In this problem a collection of agents controls the movement of a set of vehicles approaching an exit point, with the goal of coordinating the flow through this exit. We would like to solve this problem using local information only and distributed control, while avoiding collisions. This “polite” exit strategy has important applications to driving a mass of flying drones to land on a hard-to-reach target.

Our solution is a gradient descent algorithm with a suitable artificial potential function [2, 3, 4]. We then transform the classical (centralized) solution into a decentralized algorithm by restricting the “visibility” of each drone to a limited portion of the surrounding space, compatibly with the sensors capabilities.

More specifically, we describe this transformation methodology for obtaining a distributed algorithm [5, 6] and instantiate the methodology to the 3D Opera

Problem. We then characterize the analytic properties of our solutions and present extensive experiments. We implemented a Montecarlo D/E numerical simulation involving hundreds of agents, and studied the behavior under several families of potentials and a wide range of values for the parameters. In particular we studied three important families of repulsive potentials drawn from Newtonian Physics (gravity), classical Control Theory (sigmoid) and Chemistry (Lennard-Jones). Our choices were motivated by

- (a) the need for drawing from the experience of well tested approaches as those belonging to mature and classical traditions, in order to sustain the effectiveness of our methodology, and
- (b) the hope for inheriting the well established mathematical-analytical framework to be able to prove, formally, convergence and robustness properties of the algorithms in play.

2 The 3D Opera Problem

2.1 Problem Definition

We are given a 3D environment with a marked exit and a set of moving agents. Each agent in the system is modeled as a point. More formally, we are given a set of n points $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbf{R}^d$ (with $d = 2$, or $d = 3$) initially placed uniformly at random within a bounded region of \mathbf{R}^d . Each point has the goal of reaching the exit within a finite interval of time, while avoiding collisions. Each point has the ability of detecting the presence of other points only within a limited range. Our formulation captures the DARPA IVRS Ref problem [7, 8, 9]. Note that in this problem formulation the main goals are conflicting in so far as each drone needs to pass through the same aperture. This causes the agents to compete. At the same time, because agents cannot collide with each other, they also need to cooperate.

¹Research partially supported by the DARPA TASK program under grant number F30602-00-2-0585.

2.2 Artificial Potential Functions

One possible solution to the 3D Opera problem is path control using artificial potential functions. Potentials act on all the points and the absolute minimum corresponds to the goal. The agents will achieve their goal by minimizing this global potential through, for example, a gradient descent method with errors [1, 10]. Formally, let $\mathbf{x} = (x^{(1)}, \dots, x^{(d)}) \in \mathbf{R}^d$ and define $X = (\mathbf{x}_1, \dots, \mathbf{x}_n)$, $\mathbf{r}_{ij} = \mathbf{x}_i - \mathbf{x}_j$ and $r_{ij} = \|\mathbf{r}_{ij}\|$. Our potentials are a combination $F(X) = V_a(X) + V_r(X)$ of an attractive term (the main goal) and a repulsive term (safety w.r.t. collisions):

$$\begin{aligned} F(X) &= \sum_{1 \leq i \leq n} \|\mathbf{x}_i - \mathbf{0}\| + \beta \sum_{1 \leq i < j \leq n} r(\|\mathbf{x}_i - \mathbf{x}_j\|) \\ &= \sum_{1 \leq i \leq n} \|\mathbf{x}_i\| + \beta \sum_{1 \leq i < j \leq n} r(r_{ij}) \end{aligned}$$

where $r : \mathbf{R}^+ \mapsto \mathbf{R}^+$ is typically (but not necessarily) nondecreasing. In this paper we have considered the following three families of inter-drone repulsive functions:

$$\begin{aligned} \text{Gravity:} \quad r_1(x; \alpha, \eta) &= \frac{1}{x^{\alpha+\eta}}, \\ \text{Sigmoid:} \quad r_2(x; \alpha, \eta) &= \frac{1}{1+e^{\alpha(x-\eta)}}, \\ \text{Lennard-Jones:} \quad r_3(x; \alpha, \eta) &= \left(\left(\frac{\alpha}{x+\eta}\right)^6 - 1\right)\left(\frac{\alpha}{x+\eta}\right)^6, \end{aligned}$$

where $\alpha, \eta > 0$ are the characterizing parameters to be tuned together with $\beta > 0$.

3 Distributed Gradient Descent Algorithms

3.1 Distributed Potentials

Suppose the sensors of each agent have range ρ . A “distributed” version of our artificial potential function is:

$$\hat{F}_\rho(X) = \sum_{1 \leq i \leq n} \|\mathbf{x}_i\| + \beta \sum_{1 \leq i < j \leq n, r_{ij} < \rho} r(r_{ij}),$$

whose gradient $\nabla \hat{F}_\rho$ is given by

$$\nabla \hat{F}_\rho(X) = \left(\frac{\partial \hat{F}_\rho}{\partial \mathbf{x}_1}(X), \dots, \frac{\partial \hat{F}_\rho}{\partial \mathbf{x}_n}(X) \right),$$

where

$$\frac{\partial \hat{F}_\rho}{\partial \mathbf{x}_i}(X) = \left(\frac{\partial \hat{F}_\rho}{\partial x_i^{(1)}}(X), \dots, \frac{\partial \hat{F}_\rho}{\partial x_i^{(d)}}(X) \right).$$

First we observe that $\frac{\partial \|\mathbf{x}\|}{\partial \mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|}$ as follows from basic vector analysis. Then, applying the chain rule, we can express each component of ∇F , and hence of $\nabla \hat{F}_\rho$ as

$$\frac{\partial F}{\partial \mathbf{x}_i}(X) = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|} + \beta \sum_{1 \leq j \leq n, j \neq i} r'(r_{ij}) \frac{\mathbf{r}_{ij}}{r_{ij}},$$

and similarly,

$$\frac{\partial \hat{F}_\rho}{\partial \mathbf{x}_i}(X) = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|} + \beta \sum_{1 \leq j \leq n, j \neq i, r_{ij} < \rho} r'(r_{ij}) \frac{\mathbf{r}_{ij}}{r_{ij}}.$$

Finally, considering the families of potentials we mentioned before, the first derivatives of the functions r_i under investigation will be

$$\text{Gravity:} \quad r'_1(x) = -\frac{\alpha x^{\alpha-1}}{(x^{\alpha+\eta})^2},$$

$$\text{Sigmoid:} \quad r'_2(x) = -\frac{\alpha e^{\alpha(x-\eta)}}{(1+e^{\alpha(x-\eta)})^2},$$

$$\text{L.J.:} \quad r'_3(x) = \frac{6}{\alpha} \left(\frac{\alpha}{x+\eta}\right)^7 \left[1 - \frac{2}{\alpha} \left(\frac{\alpha}{x+\eta}\right)^6\right].$$

$\nabla \hat{F}_\rho$ is computable by each agent using only local information.

3.2 Gradient Descent Algorithms

The classical gradient descent method is based on the idea that each agent updates its position following the steepest decreasing direction of the potential. This is precisely the opposite of the gradient vector of the potential function. One way to implement this method is through the following rule:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) - \gamma_t \frac{\partial \hat{F}_\rho}{\partial \mathbf{x}_i}(X(t)), \quad (1)$$

for a sequence $\gamma_t > 0$ that typically satisfies $\sum \gamma_t = \infty$ and $\sum \gamma_t^2 < \infty$ [1, 10].

We have simulated this method using two algorithms. The first operates synchronously. At each round it chooses a random permutation of the agents and applies incrementally the rule (i.e., using the most recent information) to each agent in accordance to the order given by this random sequence [1]. Moreover, since our decentralized version is based on an approximate of the gradient, our method actually falls in the category of the gradient descent methods with errors, well investigated in [10].

This technique has the disadvantage that it does not take into consideration the feasibility of the kinematics necessary for the agents to reach their new computed position. In other words, it assumes implicitly

that the travel time is equal for all the agents independently upon the actual distances to be covered.

One way to circumvent this issue is to resort to a Discrete Event simulation. This corresponds to making exactly the opposite assumption: each agent travels at constant speed (in norm). In other words, each agent applies the position update rule upon reaching its current destination. Upon the occurrence of this event the agent will set new course and proceed at constant speed towards the newly computed destination.

Thus, the algorithm may be described as follows. At the beginning, say at time $t = 0$, all the agents, placed uniformly at random within their environment, compute their current destination, together with the needed travel times. This is accomplished by applying the position update rule (1) and assuming that the velocities are constant in norm (and here for simplicity equal to v for all). This first step produces:

- (a) a description of the system state that consists of the current agent positions and velocities $X = \{\mathbf{x}_i\}$ and $V = \{\mathbf{v}_i\}$;
- (b) a discrete set E of events (t, i) signifying that agent i will reach its current designated destination at time t .

Then the simulation proceeds with the extraction of the event (t, j) that is meant to occur first, i.e., the one that minimizes the element t , and finally with the update of the system state. This final step is carried out in the following way:

1. The positions $X(t)$ of all the agents at time t are computed using kinematics (linear uniform motion along the directions \mathbf{v}_i). During this step the agents that have reached their ultimate destination, the gate area, are marked exited and will no longer be considered.
2. The position update rule is applied to the agent j associated with the extracted event (t, j) to compute its new designated destination and, as a consequence, its new velocity \mathbf{v}'_j and travel time t'_j :

$$\mathbf{x}'_j = \mathbf{x}_j + \gamma_t \frac{\partial \hat{F}_\rho}{\partial \mathbf{x}_j}(X), \quad \mathbf{v}'_j = v \frac{\mathbf{x}'_j - \mathbf{x}_j}{\|\mathbf{x}'_j - \mathbf{x}_j\|}, \quad (2)$$

$$t'_j = \frac{\|\mathbf{x}'_j - \mathbf{x}_j\|}{v}.$$

This in turn causes the generation of a new event $(t + t'_j, j)$ to be included in the set E .

A pseudo-code description of our Discrete Event Simulation algorithm would be the following:

```

E ← ∅
for 1 ≤ i ≤ n do
    compute  $\mathbf{x}'_i, \mathbf{v}'_i$  and  $t'_i$  according to (2)
     $\mathbf{v}_i \leftarrow \mathbf{v}'_i$ 
    E ← E ∪ {(t'_i, i)}
endo
while E ≠ ∅ do
    (t, i) ← min(E)
    E ← E \ {(t, i)}
    update X at time t
    compute  $\mathbf{x}'_i, \mathbf{v}'_i$  and  $t'_i$  according to (2)
    E ← E ∪ {(t + t'_i, i)}
     $\mathbf{v}_i \leftarrow \mathbf{v}'_i$ 
endo

```

The algorithm is guaranteed to terminate as long as the agents will all reach the gate area within a finite interval of time.

4 Analysis

A typical weakness of navigation approaches based on the artificial potential method is represented by the possible undesirable presence of local minima of the potential function within the working domain [2, 11, 12]. This means that the system might reach a state that locally minimizes the potential.

We show, now, how to approach the issue of the local minima when the repulsive inter-drone function r have bounded first derivative r' . In those cases, we may prove formal facts about the robustness and the convergence of the system.

The general idea behind our proofs is to exploit the boundedness of r' . For example, given r , define r^* as the maximum of r' over $\mathbf{R}_{\geq 0}$. Then we can say that

$$\left\| \sum_{1 \leq j \leq n, j \neq i} r'(r_{ij}) \frac{\mathbf{r}_{ij}}{r_{ij}} \right\| \leq (n-1)\beta r^*$$

and so, for β satisfying

$$\beta \leq \frac{1}{(n-1)r^*}$$

we are guaranteed that the attraction will always prevail over the repulsion ruling out the existence of local minima.

Let R be the 3D bounded environment inside which we deploy our n drones. For simplicity we can take

$R = \{(x, y, z) \in \mathbf{R}^3 \mid |x, y| \leq l/2, 0 \leq z \leq l\}$ the parallelepiped of side l . The point $O = (0, 0, 0)$ towards which the drones will be ultimately attracted is then located at the middle of the *floor* of R . We also assume that the *gate* the drones need to penetrate through is a set of nonnull measure. For example we can consider a ball or a semiball G centered at O and of radius ρ_0 : $G = B(O, \rho_0) \cap \{(x, y, z) \mid z > 0\}$. Under this assumption it is intended that any drone crossing G is to be considered *exited*.

The results we are about to state apply to both the synchronous and asynchronous gradient descent algorithms deriving from the following artificial potential function $\hat{F}_\rho(X) = \sum_{1 \leq i \leq n} \|\mathbf{x}_i\| + \beta \sum_{1 \leq i < j \leq n, r_{ij} < \rho} r_2(r_{ij})$, where $r_2(x) = \frac{1}{1 + e^{\alpha(x-\eta)}}$. Similar results can be established for the other functions r and in general for any bounded repulsive function. The position update rule in both the algorithms will then take the following shape:

$$\mathbf{x}'_i = \mathbf{x}_i - \gamma_t \left(\frac{\mathbf{x}_i}{\|\mathbf{x}_i\|} + \beta \sum_{1 \leq j \leq n, j \neq i, r_{ij} < \rho} r'_3(r_{ij}) \frac{\mathbf{r}_{ij}}{r_{ij}} \right).$$

Theorem 1 *Let $0 < \delta < 1$, $0 < \beta < \frac{4(1-\delta)}{\alpha^2(n-1)}$, $\gamma_t = 1$, $\forall t$. Then all the drones reach the gate G within a finite interval of time.*

Proof: The idea behind the proof is that, if we are willing to lower the parameter β under a certain threshold, the attractive term of the gradient will always dominate over the repulsive one. And this would ensure that at each step any drone updating its position will have to decrease its distance from the gate of a nonnegligible (noninfinitesimal) quantity.

Since

$$\left\| \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|} \right\| = 1 \text{ and } \max_{0 \leq x \leq \rho} |r'_3(x)| = r'_3(\eta) = \frac{\alpha}{4},$$

it follows that for $\beta < \frac{4(1-\delta)}{\alpha^2(n-1)}$ the following inequality holds

$$\left\| \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|} \right\| > \delta + \alpha\beta \left\| \sum_{1 \leq j \leq n, j \neq i, r_{ij} < \rho} r'_3(r_{ij}) \frac{\mathbf{r}_{ij}}{r_{ij}} \right\|.$$

This implies that, at each recomputation of its new position, the drone in play will move closer to the gate. To conclude the proof we need to show that the measure of the radial component of $\mathbf{x}'_i - \mathbf{x}_i = -\gamma_t \cdot \frac{\partial \hat{F}_\rho}{\partial \mathbf{x}_i}(X(t))$ is lower bounded by a positive quantity. But this follows from the fact that $\gamma_t = 1$ and so $\|\mathbf{x}'_i\| \leq \|\mathbf{x}_i\| - \delta$.
□

This result does not provide any collision guarantees. In section 5 we show experimentally that collisions do not happen. For the rest of this section we extend our analysis to collisions.

At each step the system moves to a state that strictly decreases the total potential F , as prescribed by the gradient descent method. Moreover F is a summation of positive terms so that, at each time $t > 0$, each summand in $F(X(t))$ must be necessarily bounded by $F(X(t)) < F(X(0))$. In particular this must be true for the terms $r(r_{ij}(t)) = r(\|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|)$. But this implies that, at any time in the future, the inter-drone distance repulsion will never be greater than the initial value of the total potential $F(X(0))$, i.e., $\beta \max_{i,j,t} r(r_{ij}(t)) \leq F(X(0))$. Let us now define $F_0 = \sum_{1 \leq i \leq n} \|\mathbf{x}_i(0)\| + \beta \sum_{1 \leq i < j \leq n} r(r_{ij}(0)) = P_0 + \beta R_0$, and focus on a nonincreasing function r . Then we can invert r and transform the above inequality into $\min_{i,j,t} \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\| \geq r^{-1} \left(\frac{F_0}{\beta} \right) \geq r^{-1} \left(\frac{P_0}{\beta} + R_0 \right)$. From this we may observe that we could try to establish appropriate values of the parameter β in order to keep the minimum inter-drone distance above a given threshold. Nevertheless we point out that this technique, based only on varying the parameter β alone, is quite limited. In fact it does not allow us to control the magnitude of the term R_0 , that depends on the initial scattering of the agents, and worse, it hides the role of the other parameters, α, η , that affect R_0 and seem to be somehow interrelated.

As a matter of exemplifying, let us choose r to be the sigmoid function (r_2) which ranges nonincreasingly on $(0, 1)$. We can immediately prove the following lemma.

Lemma 2 *Let $\beta > 0$ and the total magnitude of the potential F_0 at time $t = 0$ be $F_0 \leq \beta/2$. Then $\min_{i,j,t} \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\| \geq \eta$.*

Proof: Since r is monotone we can determine its inverse: $r^{-1}(y) = \eta + \frac{1}{\alpha} \ln \left(\frac{1}{y} - 1 \right)$. Thus for $F_0 \leq \beta/2$ the following inequalities hold $\min_{i,j,t} \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\| \geq r^{-1} \left(\frac{F_0}{\beta} \right) \geq \eta + \frac{1}{\alpha} \ln \left(\frac{\beta}{F_0} - 1 \right) \geq \eta$.
□

The following theorem allows us to determine values of β for which the inter-drone distance never drops below a given threshold, provided that, at the beginning (time $t = 0$), we have enough space to scatter conveniently our drones.

Theorem 3 Suppose we are given n drones and let $\alpha, \eta > 0$. Assume that the initial scattering at time $t = 0$ yields $\min_{i,j} \|\mathbf{x}_i(0) - \mathbf{x}_j(0)\| \geq \eta + \frac{2}{\alpha} \ln 2n$. Then for $\beta > 8P_0$ we are guaranteed that $\min_{i,j,t} \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\| \geq \eta$.

Proof: The idea here is to show that with the given scattering and value of β the conditions of Lemma 2 are verified and that would be enough for the claim. Using the definition of R_0 and exploiting the monotonicity (nonincreasing) of r we can derive immediately the following chain of inequalities:

$R_0 \leq n(n-1) \max_{i,j} r(\|\mathbf{x}_i(0) - \mathbf{x}_j(0)\|) \leq n(n-1)r(\eta + \frac{2}{\alpha} \ln 2n) \leq \frac{n(n-1)}{1+e^{\alpha((\eta+(2/\alpha)\ln 2n)-\eta)}} \leq \frac{n(n-1)}{1+4n^2} \leq \frac{3}{8}$. Now, since $\beta \geq 8P_0$ and $R_0 \leq 3/8$ we have $F_0 = P_0 + \beta R_0 \leq \frac{\beta}{8} + \beta R_0 \leq \frac{\beta}{8} + \beta \frac{3}{8} = \frac{\beta}{2}$. But this is indeed the condition for the application of Lemma 2. \square

5 Experiments

We have run several Montecarlo simulations of a Discrete Event implementation of our distributed gradient descent algorithm.

The discrete event simulation determines a sequence of $N > 0$ time steps $\Delta_0, \Delta_1, \dots, \Delta_N$ so that the positions X (and speeds V) can change only at time t_1, t_2, \dots, t_N , where

$$t_i = \sum_{0 \leq k < i} \Delta_k.$$

As a matter of definition we set $t_0 = 0$ and $T = t_N$.

Given two points $1 \leq i < j \leq n$, we denote the minimum distance between them during the k -th time interval as $D(i, j, k) = \min_{t \in [t_{k-1}, t_k]} \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|$. Notice that the minimum distance during all the simulation can hence be computed as $\min_{t \in [0, T]} \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\| = \min_{1 \leq k \leq N} D(i, j, k)$. We also introduce the following measures:

$$D(i, k) = \min_{j \neq i} D(i, j, k), \text{ and } \bar{D}(k) = \frac{1}{n} \sum_{1 \leq i \leq n} D(i, k)$$

where we observe that $\bar{D}(k)$ depends only on the time step k .

To eliminate the dependence on k we considered two possible statistics: the (arithmetic) mean \bar{D}_{av} and the median \bar{D}_{md} of the $N + 1$ observations $\{\bar{D}(k)\}_{0 \leq k \leq N}$. The choice of the median was motivated by empirical criteria of robustness. For example we observed that $\bar{D}(k)$ has very small values at the beginning of the simulation (due to the initial random position of the points) and very large values at the end of the simulation (due to the very low density of the

remaining points) and those would undesirably affect the mean. The resort to the median was intended to compensate those effects.

Let us now derive a formula to compute $r_{i,j,k}$. The equation of motions of \mathbf{x}_i and \mathbf{x}_j for $t \in [t_{k-1}, t_k]$ are given by

$$\begin{aligned} \mathbf{x}_i(t) &= \mathbf{x}_i(t_{k-1}) + (t - t_{k-1})\mathbf{v}_i(t_{k-1}) \\ \mathbf{x}_j(t) &= \mathbf{x}_j(t_{k-1}) + (t - t_{k-1})\mathbf{v}_j(t_{k-1}) \end{aligned}$$

so that $\mathbf{r}_{ij}(t) = \mathbf{x}_i(t) - \mathbf{x}_j(t) = \Delta \mathbf{x} + s \Delta \mathbf{v}$, where we defined $\Delta \mathbf{x} = \mathbf{x}_i(t_{k-1}) - \mathbf{x}_j(t_{k-1})$, $\Delta \mathbf{v} = \mathbf{v}_i(t_{k-1}) - \mathbf{v}_j(t_{k-1})$ and $s = t - t_{k-1}$. Then,

$$D(i, j, k) = \min_{s \in [0, \Delta_k]} \|\Delta \mathbf{x} + s \Delta \mathbf{v}\|.$$

Applying again the chain rule, we have

$$\frac{d\|\Delta \mathbf{x} + s \Delta \mathbf{v}\|}{ds} = \frac{\Delta \mathbf{x} + s \Delta \mathbf{v}}{\|\Delta \mathbf{x} + s \Delta \mathbf{v}\|} \Delta \mathbf{v} = \frac{\Delta \mathbf{x} \Delta \mathbf{v} + s \|\Delta \mathbf{v}\|^2}{\|\Delta \mathbf{x} + s \Delta \mathbf{v}\|}.$$

As one can verify, such a derivative nullifies at

$$s^* = -\frac{\Delta \mathbf{x} \Delta \mathbf{v}}{\|\Delta \mathbf{v}\|^2},$$

so that

$$D(i, j, k) = \min_{s \in \{0, s^*, \Delta_k\}} \|\Delta \mathbf{x} + s \Delta \mathbf{v}\|.$$

The algorithms we described so far were implemented in Matlab and C. We collected data in the form of groups of tables that report the average over 100 runs of the quantities \bar{D}_{av} (average mean), \bar{D}_{md} (average median), N and T . In each trial we have placed the points, uniformly at random, within a cube of side 10 and assumed that the sensors range ρ was 1.5 (the goal is considered achieved by point i when $\|\mathbf{x}_i\| \leq 0.5$). Due to space constraints we include here only the data regarding the Lennard-Jones potential (see table 1, below) displaying the best overall performances. The highlighted values show the best behavior of the algorithm with respect to the parameter setting.

6 Conclusions

The experimental data reveals that the apparently conflicting metrics \bar{D}_{av} and N are *nonintuitively* optimized at the same time for specific choices of the parameters.

We may also observe that β acts as an evident amplification factor determining the weight of the repulsion V_r within the total expression F , α controls the ‘‘compression’’ of the argument and η controls the horizontal translation of the function.

Thus, we expect all the metrics to be sensitive to the parameter η , even suggesting the existence of critical values. This, because mild translations of r_i correspond to high rate decreases or increases of its least upper bound around $x = 0$.

Finally, let us notice that the agents seem to maintain a reasonable interdistance although the values of β are not particularly high as the theory would suggest.

References

- [1] Valentino Crespì, George Cybenko, and Daniela Rus. Decentralized Control and Agent-Based Systems in the Framework of the IRVS. *Darpa Task Program paper*. <http://actcomm.thayer.dartmouth.edu/task/>, Apr 2001.
- [2] J.C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- [3] Yong K. Hwang and Narendra Ahuja. Gross Motion Planning – A survey. *ACM Computing Surveys*, 24(3):219–291, September 1992.
- [4] Elon Rimon and Daniel E. Koditschek. Exact Robot Navigation using Artificial Potential Functions. *IEEE Transactions on Robotics and Automation*, October 1992, 8(5):501–518, 1992.
- [5] George Cybenko. Agent-Based Systems Engineering. *Darpa Task Program Research Proposal*. <http://actcomm.thayer.dartmouth.edu/task/>, Oct 2000.
- [6] Valentino Crespì and George Cybenko. Agent-based Systems Engineering and Intelligent Vehicles and Road Systems. *Darpa Task Program white paper*. <http://actcomm.thayer.dartmouth.edu/task/>, April 2001.
- [7] T. Frazier and A. Newman. Intelligent Roadway Vehicle System. *REF for the DARPA ISO TASK Program*, Oct 2000.
- [8] T. Frazier, A. Newman, and L. Roszman. IVRS REF Scenarios. DARPA Progress Report. *Darpa ISO Task Report*, December 2000.
- [9] T. Frazier, A. Newman, and L. Roszman. CAHDE REF Scenarios. DARPA progress report. *Darpa ISO Task Report*, April 2001.
- [10] D.P. Bertsekas and J.N. Tsitsiklis. Gradient Convergence in Gradient Methods. *SIAM J. on Optimization*, 10:627–642, 2000.

- [11] Hans Jacob S. Feder and Jean-Jacques E. Slotine. Real-Time Path Planning using Harmonic Potentials in Dynamic Environments. In *Proceedings of the International Conference on Robotics and Automation*, pages 874–881, 1997.
- [12] Leena Singh, Herry Stephanou, and John Wen. Real-time Robot Motion Control with Circulatory Fields. In *Proceedings of the International Conference on Robotics and Automation*, pages 2737–2742, 1996.

| β | α | η | \bar{D}_{av} | \bar{D}_{md} | N | T |
|---------|----------|--------|----------------|----------------|------------|--------------|
| 0.9 | 0.25 | 0.9 | 0.790 | 0.781 | 575 | 11.00 |
| 0.9 | 0.25 | 1.0 | 0.791 | 0.794 | 574 | 11.00 |
| 0.9 | 0.25 | 1.1 | 0.791 | 0.801 | 571 | 11.00 |
| 0.9 | 0.5 | 0.9 | 0.777 | 0.785 | 570 | 11.01 |
| 0.9 | 0.5 | 1.0 | 0.786 | 0.791 | 571 | 11.01 |
| 0.9 | 0.5 | 1.1 | 0.789 | 0.782 | 573 | 11.00 |
| 0.9 | 0.75 | 0.9 | 0.754 | 0.748 | 577 | 11.70 |
| 0.9 | 0.75 | 1.0 | 0.763 | 0.750 | 574 | 11.13 |
| 0.9 | 0.75 | 1.1 | 0.769 | 0.786 | 569 | 11.05 |
| 1.0 | 0.25 | 0.9 | 0.792 | 0.787 | 574 | 11.00 |
| 1.0 | 0.25 | 1.0 | 0.800 | 0.803 | 572 | 11.00 |
| 1.0 | 0.25 | 1.1 | 0.794 | 0.807 | 574 | 11.00 |
| 1.0 | 0.5 | 0.9 | 0.793 | 0.813 | 567 | 11.01 |
| 1.0 | 0.5 | 1.0 | 0.801 | 0.818 | 568 | 11.01 |
| 1.0 | 0.5 | 1.1 | 0.795 | 0.791 | 573 | 11.00 |
| 1.0 | 0.75 | 0.9 | 0.753 | 0.738 | 583 | 12.16 |
| 1.0 | 0.75 | 1.0 | 0.763 | 0.754 | 573 | 11.14 |
| 1.0 | 0.75 | 1.1 | 0.767 | 0.778 | 568 | 11.06 |
| 1.1 | 0.25 | 0.9 | 0.795 | 0.790 | 574 | 11.00 |
| 1.1 | 0.25 | 1.0 | 0.799 | 0.783 | 572 | 11.00 |
| 1.1 | 0.25 | 1.1 | 0.792 | 0.799 | 573 | 11.00 |
| 1.1 | 0.5 | 0.9 | 0.790 | 0.786 | 572 | 11.01 |
| 1.1 | 0.5 | 1.0 | 0.789 | 0.797 | 567 | 11.01 |
| 1.1 | 0.5 | 1.1 | 0.798 | 0.799 | 566 | 11.00 |
| 1.1 | 0.75 | 0.9 | 0.738 | 0.727 | 584 | 12.38 |
| 1.1 | 0.75 | 1.0 | 0.753 | 0.745 | 575 | 11.38 |
| 1.1 | 0.75 | 1.1 | 0.767 | 0.779 | 567 | 11.10 |

Table 1: L.J. potential with $\gamma_t = 1.0$ and $n = 100$.