An exploratory study in formation optimization

Qi Ma
Department of Mechanical and Process Engineering
ETHz
Zurich, Switzerland
qimaqi@student.ethz.ch

Hudoba de Badyn Mathias
Department of Information Technology and Electrical Engineering
ETHz
Zurich, Switzerland
mbadyn@ethz.ch

Florian Dörfler
Department of Information Technology and Electrical Engineering
ETHz
Zurich, Switzerland
doerfler@control.ee.ethz.ch

Wenjun Mei
Department of Information Technology and Electrical Engineering
ETHz
Zurich, Switzerland
meiwenjunbd@gmail.com

***Abstract*—This paper presents a formation optimization method based on finite-state machine and decentralized MPC. By attributing different cost items into state machine, the agent can perform certain intelligent behavior like obstacle avoidance, predator avoidance and younger protection. These behaviors are demonstrated by numerical simulation, where the effect of time horizon, agent number are analyzed. Our work provides a solution for optimizing the best path for the multi-agent system with different weights in the presence of hazards and obstacles.**

Keywords—formation control and optimization, finite-state machine, decentralized MPC

I. INTRODUCTION

Flocking behavior can be observed in the migrations of animal groups like wildebeest. By interacting with local neighbors each agent inside the group will adjust its position and velocity to form a certain formation for the common group objective: predators repelling, resources sharing and young raising[1].

In recent years the multi-agent systems based on rich biological examples raises tremendous research interests. Reynolds proposed the first flocking algorithm with three heuristic rules[2]: (1) Flocking cohesion, (2) Flocking separation, (3) Flocking alignment. Vicsek studied the alignment of flocking behavior by calculating the average direction of neighbors[3]. Work [4] summarizes the flocking-related theory and algorithms by constructing the framework with three classes of flock members. It provides the collective potential for keeping the lattice-shape.

By formulating the formation control problem into solving the minimum of cost function make it possible for multiple intelligent behaviors: obstacle avoidance, target tracking[4]. Considering the predictive intelligence of animals swarms. Zhang[5] introduces the predictive mechanism into multi-agent systems and shows the importance of predictive mechanisms to multi-agent consensus.

Although plenty of research was conducted for formation optimization it focuses mainly on avoiding fragmentation and achieving a high convergence rate with low lattice regularity[6]. In this work we analyze the formation optimization problem of wildebeest migrations and show its superiority compared to other formation strategies. To better mimic the flocking behavior of wildebeest, we treat agents with different importance (young, middle-age and old-age). By adopting finite machine states our multi-agent system shows animal-flocking-like behavior including target tracking, obstacle avoidance, predator escape and important-agents protection.

The rest of paper is organized as follows. The problem formulation and modeling are presented in section 2. The behavior-based discrete-time MPC for the multi-agent system is provided in section 3. In section 4 we show our simulation and results. Finally, we make our conclusion in section 5.

II. PROBLEM FORMULATION

A. Formation dynamics

First, a standard multi-agent system with N agents with dynamics as follows:

$$\begin{cases} q_i(k+1) = q_i(k) + Ts * p_i(k) \\ p_i(k+1) = p_i(k) + Ts * u_i(k) \end{cases} \tag{1}$$

The $q_i(k), p_i(k)$ denotes the position and velocity of agent *i* and $u_i(k)$ denotes the acceleration of the control input to be optimized. Since for animals like wildebeest the pose of each agent should not be neglected and the combination of linear acceleration with angle acceleration is more animal-like than velocity in *x* and *y* direction. So we reformulate the dynamics as following.

$$\begin{cases} x_i(k+1) = x_i(k) + Ts * v_i(k) * \cos(\theta_i) \\ y_i(k+1) = y_i(k) + Ts * v_i(k) * \sin(\theta_i) \\ \theta_i(k+1) = \theta_i(k) + Ts * \omega_i(k) \\ v_i(k+1) = v_i(k) + Ts * \alpha_i(k) \\ \omega_i(k+1) = w_i(k) + Ts * \beta_i(k) \end{cases} \tag{2}$$

Where the x_i, y_i, θ_i denotes the position and orientation of agent i on the 2D world. The v_i, ω_i represent the linear velocity and angular velocity while the α_i, β_i denotes the linear acceleration and angular acceleration. As Fig 1 shows, the blue circle represents the agent and the ray leading from the center of the circle indicates the heading direction. Moreover, each agent associate with a certain weight to indicates its importance (young, middle-age, old).

Then we analyzed the characteristics of wildebeest migrations and modeled these characteristics in conjunction.

B. Wildebeest migrations modelling

The wildebeest migrate in a long range in search of water and grass. This behavior is cyclical and the migration paths remain relatively constant. Therefore we model the migrations problem by setting the tracking task of agents from start point to end point.

As shown in Fig 1, the red cross at the bottom left is the start point of the migration while the one at top right represents the end point of the migration.

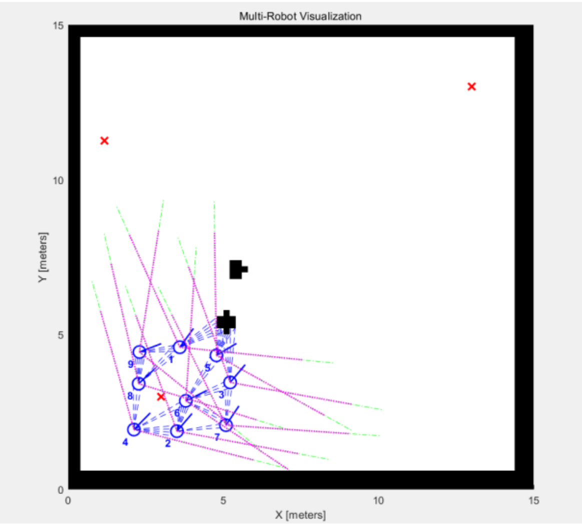


Fig. 1. Modelling of Wildebeest migrations with matlab multi-robot toolbox[7].Blue circles are agents, black object in center is obstacles. The red cross in the bottom left and top right indicates the start point and end point. The cross in the left top represent the preadtor.

The primary goal of this work is to design the control law $\alpha_i(k), \beta_i(k)$ for each agent with local information available. The control laws are optimized to helps the distributed multi-agent system reach the endpoint as quickly as possible with the least loss of their group. We firstly introduce some basic concepts in graph theory.

The detection field of a wildebeest is a conical area so except for the face-to-face situation (rare when migrating) the wildebeest cannot get connection with each other so the local information like range, importance and velocity are not shared. We build the directed graph $G = (V, \mathrm{E})$ to present the formation connection. The vertex set $V = \{1,2,\dots,n\}$ denotes the agents and set $\mathrm{E} \subseteq V \times V = \{(i,j)|i,j \in V\}$ represent the directed edges between agent i and agents it detected. By define the detection maximal range d and field of view f we can define the detection field D_i of node i:

$$D_i(k) = \{j \in V\colon |\arccos(h_i \cdot h_{ij})| \le f,$$

$$\sqrt{\left(\left(x_i(k) - x_j(k)\right)^2 + \left(y_i(k) - y_j(k)\right)^2\right)} \le d\} \qquad (3)$$

The h_i, h_{ij} denotes the normalized vector of the heading of agent i and edge from node i to node j. The purple dash line shown in Fig 1 represent the detection field of each agent. The blue dash line indicates the range information is perceived by the agent behind.

After that we can analyze the characteristics of wildebeest migrations and model with mathematical description.

- In the migration the objective for all agents is to arrive the end point alive.
- In the migration one agent should keep a safe distance from the agent in front. Not too far and also not too close.
- In the migration the agent should maintain similar line speed to the previous agent to avoid collisions.
- In the migration the agent will tend to be close to important individuals (young) to protect their safety.
- In the migration the important agent will tend to approach the whole group for shelter.
- In the migration the predator(lion) will try to hunt down the closest agents.
- When the agent (wildebeest) senses out the predator(lion) is around it will focus on escaping.
- When the agent (wildebeest) detects the obstacles it will focus on more on the avoiding them.
- The agents are safe in the end point.

From above characteristics we reorganize the interaction of neighbors and its behaviors in wildebeest migratory into following laws.

1. **Flocking cohesion and separation laws:** Inspired by the work[4][6] the cohesion and separation can be achieved by designing the collective potential functions to finish the local tasks:

$$||q_j(k) - q_i(k)|| = d, \forall j \in N_i(k)$$

Where d is the desired distance in a quasi-α formation. By formulating the task into energy form we get the first item in cost function.

$$J1_i = \sum_{j \in N_i(k)} w_1 \left(||\sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} - d||^2 \right) \quad (4)$$

By minimizing this term the relative distance between the agent and agents in front will be ensured.

2. **Flocking alignment laws**: The flocking alignment keeps the speed of agent i consistent with the front one. The consensus can be achieved by alignment and the agents go line up with agent in front.

$$J2_i = \sum_{j \in N_i(k)} w_2 \left(|v_i - v_j| \right) \qquad (5)$$

By minimizing this term the linear velocity of agent i goes consensus with other agents.

3. **Flocking obstacle avoidance laws:** The flocking algorithm with obstacle avoidance create a new species of agents called β agents.

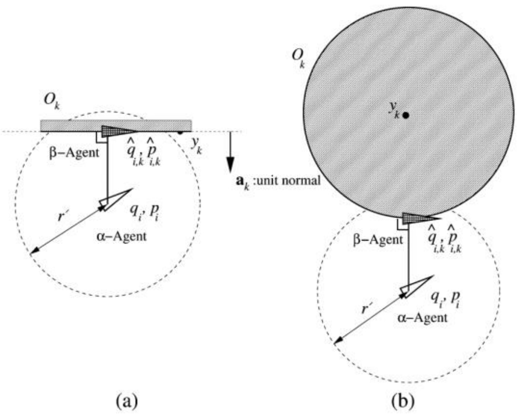


Fig. 2. Agent-based representation of obstacles. (a) Wall. (b). Spherical obstacle

To avoid the obstacles we create the term to penalize the difference of velocity p_i and $\widehat{p_{i,k}}$. By minimizing this term the agent will avoid the obstacle with best direction in parallel to the obstacle. In our work only the wall obstacles are considered.

$$J3_i = \sum_{k\in N_i^{\beta}(k)} w_3 \| v_i(\begin{bmatrix} cos\theta_i \\ sin\theta_i \end{bmatrix} - \frac{\widehat{p_{i,k}}}{|p_{i,k}|}) \| \qquad (6)$$

The $\frac{\widehat{p_{i,k}}}{|p_{i,k}|}$ indicates the velocity vector in the direction of $\widehat{p_{i,k}}$ with same linear velocity as v_i. Moreover, in order to smoothly avoid obstacles a potential weight is adopted.

$$W_{3,i} = \frac{1}{0.01 + \|q_i - \widehat{q_{i,k}}\|} \qquad (7)$$

With a dynamic weight the obstacle avoidance term will take on more weights in cost function.

4. **Flocking predator avoidance law:** Similar to obstacle avoidance the predator avoidance penalize the current direction and the best escape direction $\widehat{p_{i,e}}$.

$$J4_i = w_{4,i} \| v_{upper} \left(\begin{bmatrix} cos\theta_i \\ sin\theta_i \end{bmatrix} - \frac{\widehat{p_{i,e}}}{|p_{i,e}|} \right) \| \qquad (8)$$

In our work only one predator is considered and the escape linear velocity is set to the maximum velocity. Similarly, the weights will increase with the closer distance between the agent and predator.

5. **Flocking protection law:** Less important agent(older) will come closer to the important agent (young) and provide protection. The importance of agent j in neighbor N_i is available to agent i. So the term is designed for bringing the agent i to agent j.

$$J5_i = \sum_{j\in N_i(k)} w_{5,i} 10^{W_i - W_j} \left(\sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \right) \quad (9)$$

The W_i, W_j indicate the importance of agent i and agent j. By minimizing this term the agents with large deviation in importance will come closer with each other. This behavior is highly consistent with young animal protection in animal kingdom.

6. **Flocking target tracking law:** The most important objective for the flocking is to arrive the end point as soon as possible so we add the energy costing term for the stage cost.

$$J6_i = w_{6,i} \left(\sqrt{\left(x_i - x_{goal}\right)^2 + \left(y_i - y_{goal}\right)^2} \right)$$

$$+ w_{7,i} \frac{1}{2} \alpha^2 + w_{8,i} \frac{1}{2} \beta^2 \qquad (10)$$

By minimizing this term above the agent will go with the trajectory which spend the minimum energy in linear acceleration and angular acceleration.

In summary we model the wildebeest behavior with multiple cost function terms and the best control input can be determined by finding the minimum of the cost function. However, many terms are contradictory to each other, such as obstacles avoiding and predators avoiding. In next section we propose a behavior-based controller to connect each cost function term differently.

III. Behavior-based discrete-Time MPC

In this section a decentralized Model Predictive Controller is proposed based on flocking protocol. In order to better mimic the behavior of wildebeest we adopt the behavior-based method. Similar to animal-like thinking that there is always a main task which owns highest priority: Foraging, courtship, predators escaping, we divide the behavior of agent (wildebeest) in our work into 3 categories: Free moving, obstacle avoidance and predator avoidance as shown in Fig 3.

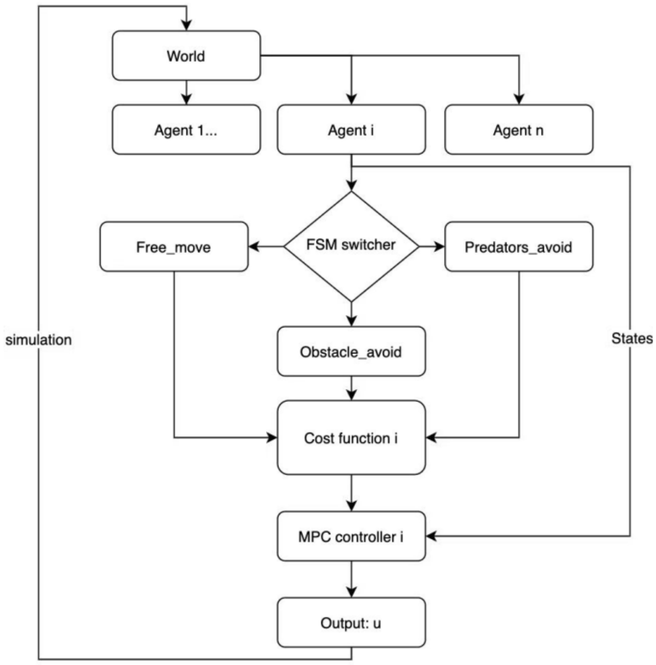


Fig. 3. Baseline of our method with finite state machine: Free move, obstacle avoidance and predators avoidance combined with MPC controller.

By adopting finite-state machine method we add the items 1,2,5,6 into cost function for free moving. The cost items 1,2,3,5,6 into the obstacle avoidance case and the items 4,5,6 for predators avoidance. In predator escaping case the agent is allowed to move away from the flocking just like the actual situation in nature.

Inspired from work[6] and equation (2) we formulate the control problem for decentralized MPC as following.

$$X_i(k) = col\left(x_i(k), y_i(k), \theta_i(k), v_i(k), \omega_i(k)\right)$$

$$U_i(k) = col(\alpha, \beta)$$

$$X_i(k+1) = A_i X_i(k) + B_i U_i(k) \qquad (11)$$

Where

$$A_i = \begin{bmatrix} 1 & 0 & 0 & Tscos(\theta(k)) & 0 \\ 0 & 1 & 0 & Tssin(\theta(k)) & 0 \\ 0 & 0 & 1 & 0 & Ts \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B_i = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ Ts & 0 \\ 0 & Ts \end{bmatrix} \tag{12}$$

However since the heading of agent and linear acceleration and angular acceleration is considered, the dynamics is nonlinear so we use nlmpc[8] from matlab to build the MPC controller.

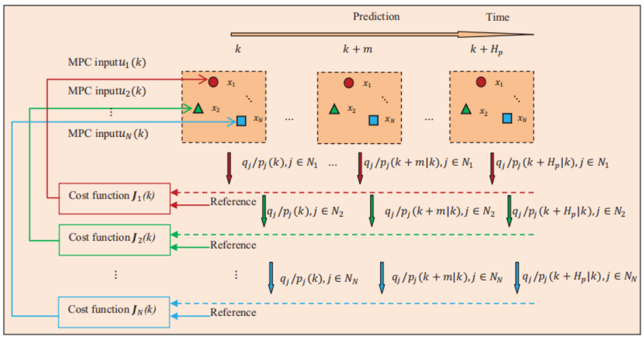


Fig. 4. Decentralized model predictive controller with local prediction[6].

As Fig 4 shows every agent will decide its optimal control input based on the information from neighbors. In a given timestamps, each agent sequentially solves their optimization problem one after the other. Firstly we assume that there are no control input for all neighbor agents. Once problem is solved for agent i, we get the optimal control input in the prediction time horizon and we update the prediction states for I and use it as reference to solve optimization problem for other agent j. Furthermore, since transition may occur in the future and the prediction become less reliable as time goes on. A decay factor is introduced to weaken the impact of future costs. The problem is written as following.

$$\begin{aligned} \min_{U_i(k)} J_{i,total}(k) &= \sum_{t=1}^{H_p} J_{i,stage} \times D^t \\ s.t.\ X_i(k+1) &= A_i X_i(k) + B_i U_i(k) \\ 0 \le v_i(k) &\le v_{upper} \\ |\omega_i(k)| &\le \omega_{upper} \\ |\alpha| &\le \alpha_{upper} \\ |\beta| &\le \beta_{upper} \end{aligned} \tag{13}$$

Where the H_p is the prediction time horizon and the decay factor $D < 1$ attach different importance to the prediction. The stage cost consists of cost items from equation 4-10.

The constrains for the states is the maximum linear velocity and angular velocity. The constrains for input is the maximum and minimum linear acceleration and angular acceleration. Since for a wildebeest moving backward is forbidden so the minimum speed is set to be 0. After optimization only the first control input is implemented to the multi-agent system.

IV. Simulation and Result

In this section we first present the simulation result to demonstrate different behavior in finite-state machine. A quantitative study is then conducted to show the performance of our system with different MPC time horizon and different robot numbers.

The simulation shows 2D multi-agent migration by using multi-robot toolbox. The blue circle shows the agent position while the blue ray indicate the heading direction of each wildebeest. All agents start randomly around start point with coordinate (3,3) and aim at moving to point with coordinate (13,13) as the two red cross shows. The moving cross start at (1,13) and represent the predator. The predator moves faster than the agent so at least one agent will be killed and the sum of importance value will less than the beginning. The goal of our work is to prove the validity of our multi-agent system which adopted the wisdom of nature.

A. *Qualitative analysis*

- Free moving state

In this state the agent will move towards to the target without worrying the obstacles or predator. A relative stable formation will be formed as Fig 5 shows. The formation is consistent with the α-lattice formation [4] [6] since we add similar collective potential into cost function.

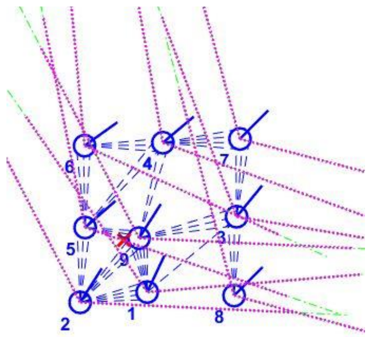


Fig. 5. Stable formation exhibits α- lattice like characteristics [6].

The importance value attach to the agent 1-9 is [1,1,1,3,3,3,3,6,6]. Higher value indicates more important. The distribution is consistent with the nature situation.(The middle-age is at most while the young is less). As the Fig 5 shows the distance between agents to agent 9. This protection behavior demonstrate the success of our multi-agent system because it will reduce the number of important members being killed.

- Obstacle avoidance state

In obstacle avoidance state the stable formation will be changed since it no longer hold the minimum potential in the cost function energy point of view. However, as the Fig 7 shows, the characteristics of the wildebeest are still preserved. Distance between the agent keep relative stable.

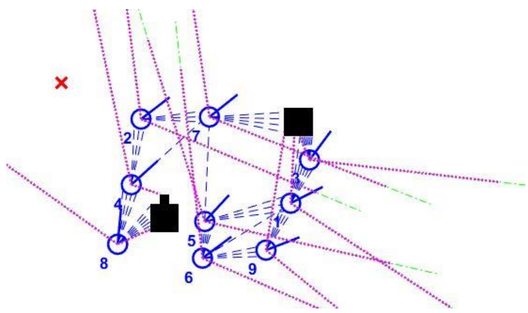


Fig. 6. Obstacle avoidance state, black square indicate the obstacles.

For later quantitative experiment two random obstacle is generated and since predator in nature world like lions use obstacles such as grasses as cover so in our work lion can cross the obstacles.

- Predator avoidance state

In formation optimization problem which aims at showing the superiority of wildebeest flocking, the predator should be considered because it also show the performance of young protection algorithm. As Fig 7 shows when the predator approach close enough to the agents. The whole flocking will spread out to avoid the attack. The agent 8 and 9 are close to other agents so it suffers lower possibility of being caught and killed.

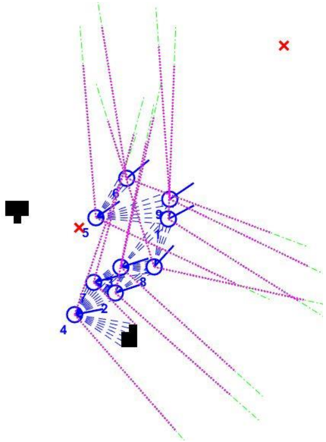
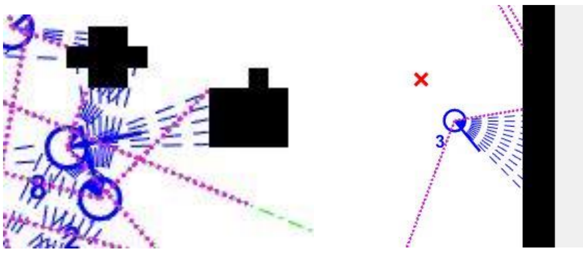


Fig. 7. Predator avoidance state, red cross in the middle indicates the predator.

After showing the different behavior of our multi-agent system we will discuss two failure cases in our early work.

- Local minima failure



(a）　　　　(b)

Fig. 8. Agent stuck at the local minima point and stand still.

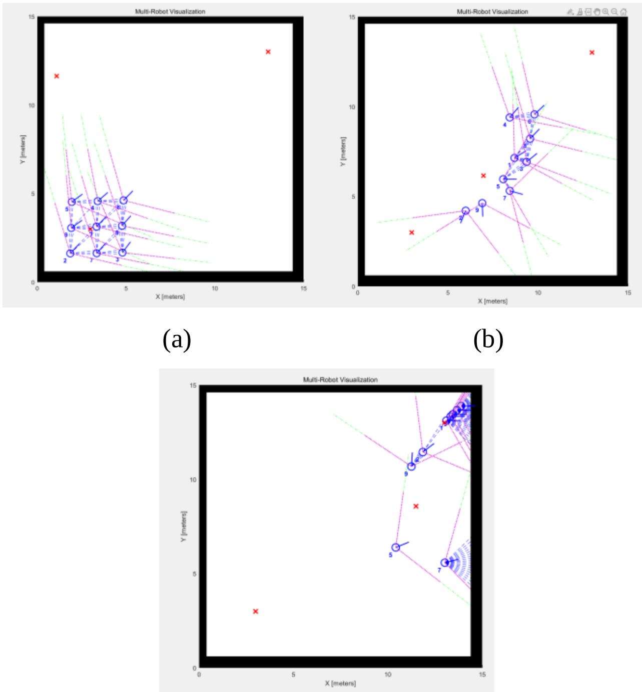
Since all control input is optimized the nonlinear cost function using fmincon[9] from matlab. This method use numerical calculation to calculate the gradient until objective function is non-decreasing in feasible directions.

As Fig 8 (a) shows two agents have opposite rotate direction in front of two very close obstacles. The costs reduction of avoiding the wall is offset by the separation costs increase. The (b) shows similar case when the costs reduction of avoiding predator is offset by the obstacle avoidance.

In order to solve this local minimum problem and inspired by roughing method in MPC controller we add a roughing noise to the pose states $(x(k), y(k), \theta(k))$ and the local minima problem is alleviated.

In summary, the consecutive snapshots of flocking are shown as Fig 9. The migration face basically three stages.

1. **Formation forming stage:** In this stage the agents will form a relative stable formation based on its initial position.
2. **Predator / Obstacle avoidance stage:** The main challenge facing by wildebeest flocking, the predator(lion) attack will decrease the sum of importance and the obstacles will slow the formation down.
3. **Arrival stage:** Influenced by the obstacle or predator the fragmentation occurs in the formation. The cohesion which leads agent to right direction and the protection for important agent is vanished. The fragmented flocking group will arrive at end point or be eaten by predator.



(a)　　　　(b)

(c)

Fig. 9. Agent stuck at the local minima point and stand still.The illustration of 2-D flocking at first, middle and end timestamps.

B. Quantitative analysis

Finally, after analyzing the qualitative studies, we conducted a quantitative study, and three metrics is used as following.

$$\bar{Q} = \sum_{i \in N_{arrival}} \frac{value_i}{|N_{arrival}|}$$

$$\bar{T} = \sum_{i \in N_{arrival}} \frac{T_{arrival}}{|N_{arrival}|}$$

$$P = \frac{\bar{Q}}{\bar{T}} \times 100 \tag{14}$$

Where the $N_{arrival}$ is the arrival set include all agents, which arrive the end point alive. The performance P depends on the sum of all values and sum of time steps of arrival.

Firstly, we would like to test the necessity of using decentralized MPC. Since MPC can not predict the transition of finite-state machine resulting in poor prediction for far future. So in following experiment we only analyze the average time cost for free moving and obstacles avoidance case. The time horizon is chosen from 1 to maxima 20 times sample time 0.05s.

As it shows in the table I in both free moving case or obstacle avoidance case the efficiency increases as time horizon increase. Moreover, we can see that the time costs for obstacle avoidance is always higher than the free moving. This is obviously due to the fact that obstacle avoidance takes more time.

TABLE I. TIME COST OF DIFFERENT MPC HORIZON

Time Horizon	**Average time cost (Ts)**	
	Free moving	**Obstacles avoid**
1	142.25	150.9
5	141.58	142.2
10	140.5	141.1
15	138.7	139.2
20	138.2	138.7

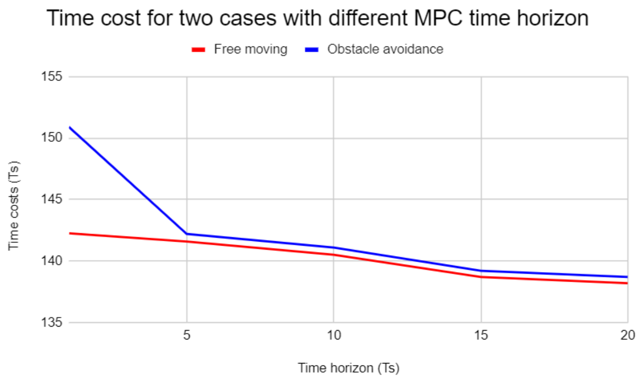


Fig. 10. The time costs change with different MPC time horizon

As Fig 10 shows by increasing the time horizon the time cost for obstacle has a very significant decline. This is because prediction is very important and helps the agent avoid rotating too much.

Secondly, we analyze the overall performance of our multi-agent system with different agent number. Since the initial position and obstacle position are random, all experiments are conducted 10 times and take the average of value sum and time costs.

As the table II shows, the overall performance increase with more agents. This can be understood from two perspectives. (1) From the basic experience with more agents the predator will face more difficult. For example, when a lion is chasing a certain agent i, its attention may be distracted by other agents especially when the agent number is high. (2) From the energy perspective when the agent number is high the cohesion, alignment and separation cost term have larger proportion of the overall cost function therefore the formation owns higher stability when facing obstacles and predator.

When the agent number is too high and the whole map will be filled with agents especially when facing the obstacles. This explains the disagreement that performance drops in the existence of obstacles but for free moving the performance is improved with increasing agent number.

TABLE II. PERFORMANCE CHANGE WITH DIFFERENT AGENT NUMBER

Agents number	**obstacles**	**average value**	**average time cost**	**Performance**
9	w	3.17	142.2	2.23
	-	2.93	141.8	2.06
18	w	3.21	146.8	2.18
	-	3.02	132.3	2.28
27	w	3.16	143.84	2.19
	-	3.08	131.4	2.34

V. CONCLUSION

To summarize this paper, we inspired from the wildebeest flocking migration and propose a multi-agent system using decentralized model predictive controller combined with finite-state machine to simulate the wildebeest migration. Intelligent flocking behaviors like obstacle avoidance, predator escaping, and young protection are achieved by adding correspondent cost items into the cost function. The qualitative and quantitative experiments justify the validity of our system which can lead agents to move to end point efficiently and robustly.

REFERENCES

[1] Clutton‐Brock, T. H., et al. "Predation, group size and mortality in a cooperative mongoose, Suricata suricatta." Journal of Animal Ecology 68.4 (1999): 672-683.

[2] Reynolds, Craig W. "Flocks, herds and schools: A distributed behavioral model." Proceedings of the 14th annual conference on Computer graphics and interactive techniques. 1987.

[3] Vicsek, Tamás, et al. "Novel type of phase transition in a system of self-driven particles." Physical review letters 75.6 (1995): 1226.

[4] Olfati-Saber, Reza. "Flocking for multi-agent dynamic systems: Algorithms and theory." IEEE Transactions on automatic control 51.3 (2006): 401-420.

[5] Zhang, Hai-Tao, et al. "Collective behavior coordination with predictive mechanisms." IEEE Circuits and Systems Magazine 8.3 (2008): 67-85.

[6] Zhan, Jingyuan, and Xiang Li. "Decentralized flocking protocol of multi-agent systems with predictive mechanisms." Proceedings of the 30th Chinese Control Conference. IEEE, 2011.

[7] Matlab Mobile Robotics Simulation Toolbox: https://www.mathworks.com/matlabcentral/fileexchange/66586-mobile-robotics-simulation-toolbox

[8] Matlab Nonlinear model predictive controller: https://www.mathworks.com/help/mpc/ref/nlmpc.html

[9] Matlab Find minimum of constrained nonlinear multivariable function: https://www.mathworks.com/help/optim/ug/fmincon.html