Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

A Continuous-time Markov Decision Process Based Method on Pursuit-Evasion Problem

Jia Shengde * **Wang Xiangke** * **Ji Xiaoting** * **Zhu Huayong** *

* *College of Mechantronic Engineering and Automation, National University of Defense Technology, Changsha, China (e-mail: jia.shde@gmail.com,xkwang@nudt.edu.cn,xiaotji@nudt.edu.cn).*

Abstract: This paper presents a method to address the pursuit-evasion problem which incorporates the behaviors of the opponent, in which a continuous-time Markov decision process (CTMDP) model is introduced, where the significant difference from Markov decision process (MDP) is that the influence of the transition time between the states is taken into account. By introducing the concept of situation, the probabilities addressing average behaviors are obtained. Furthermore, these probabilities are introduced to construct the transition matrix in the CTMDP. A policy iteration method for solving the CTMDP is also given. To demonstrate the CTMDP method for pursuit-evasion, examples in a grid environment are computed. The CTMDP-based method presented in this paper offers a new approach to pursuit-evasion modeling and may be extended to similar problems in the sequential decision process.

Keywords: Pursuit-Evasion, Continuous-time Markov Decision Process, Transition Rates Matrix, Dynamic Programming, Policy Iteration.

1. INTRODUCTION

Pursuit-evasion is a family of problems in the control theory and computer science, in which one group attempts to track down the members of another group in an environment. Due to the various applications of the ground mobile robots and unmanned aerial vehicles and so on, the existing literature on pursuit-evasion is vast in volume [Isler and et al., 2005, Virtanen et al., 2004]. The problem is difficult and usually considered as a dynamic, stochastic, continuous-space, continuous-time or discrete-time discrete-space game [Shedied, 2002].

The optimal control technique provides an efficient tool for the analysis of pursuers decisions when the dynamics of the pursuer are known. The main advantage of this method is that if a real-time solution exists, then it can be obtained by a set of difference equations [LaValle, 2006]. The optimal control technique is always used to address the problem with continuous-time and continuous-space, but the used differential equations may not actually represent the complex behaviors of the players and the solution will not be feasible while uncertainty exists in opponents and the environment. As the discretization of time and space represents another perspective, dynamic programming [Bertsekas and Tsitsiklis, 1996] theories provide a common architecture, and this discrete form suits more the calculation on computers. Through constructing a value function to be optimized and a space of states, there are two major approaches of solution well known as value iteration and policy iteration. In order to deal with the uncertainties as well as dynamics in pursuit-evasion, the dynamic programming-based techniques need to be extended. These methods that have emphasized the probabilities of addressing average behaviors will be a good choice. The description of the evaders behavior allows incorporating probabilities in opponent locations, intents and/or sensor observations.

If the probability between the states has been defined, assuming they obey the property of Markovian, the dynamic programming technique becomes a Markov decision process. The MDP is now widely accepted as a preferred frame for decision-theoretic planning [LaValle, 2006, Russell and Norvig, 2010]. Some researches using fuzzy reinforcement learning are mentioned in [Faiya et al., 2012]. It is always considered that in the context of MDP, only the states have an impact on the transition probabilities and the expected reward function, but the transition time between states is not considered. However, in a pursuit-evasion process, it is obvious that the combating result will be sensitive to time, i.e., a shorter transition time span and a longer one might lead to different results. To address this issue, the CTMDP will provide a more natural model. A brief description of the CTMDP method will be presented in Section 2 as it will be used to model an air combat strategy process.

The rest of the paper is organized as follows. Section 2 gives a brief description of the CTMDP, in which the definition of a CTMDP model and a policy algorithm of solution are presented. The dynamics of the pursuer and the evader are presented in Section 3. Section 4 describes the main contributions of this paper, in which the behavior of the opponent is modeled and a novel method for obtaining the transition rates matrix is presented. The presented method is demonstrated by some numerical examples in Section 5. And finally, concluding remarks are given in Section 6.

2. THE CONTINUOUS-TIME MARKOV DECISION PROCESS

A CTMDP model can be presented as a five tuple:

$$\{S, S_0, (A(i), i \in S), q(j|i,a), r(i,a)\}$$

The state space S is a finite set of fully observable states of the system, S_0 represents the initial state that belongs to S, $A(i)$ denotes a family of measurable subsets of actions applicable in $i \in S$, $q(j|i,a)$ is the transition rate of State j after performing action $a \in A(i)$ in State i, which can satisfy $q(j|i,a) \geq 0$ and $i \neq j$, and $r(i,a) \in R$ is a reward function such that $r(i,a)$ becomes the immediate reward for being in State i with Action a. In addition, two most important assumptions shall be kept in mind for this case. Firstly, the transition rate is conservative, i.e.

$$\sum_{j \in S} q(j|i,a) = 0. \tag{1}$$

Secondly, there is a stable condition that can be expressed as:

$$sup_{a \in A(i)} q(i|i,a) < \infty, \forall i \in S.$$

With these assumptions, the unique solution of the probability matrix for the CTMDP can be generated [Guo, 2009]. Furthermore, the transition probability matrix and the transition rates matrix satisfy the Kolmogorovs forward and backward equations [Stroock, 2005]:

$$\frac{dP(t;\mu)}{dt} = P(t;\mu)Q(\mu) \quad and \quad \frac{dP(t;\mu)}{dt} = Q(\mu)P(t;\mu) \tag{2}$$

The policy μ consists of all the actions which had been selected in every State i, i.e. $\mu = \{a_1, a_2, ..., a_{|S|}\}$, where $a_i \in A(i)$ and $|S|$ represents the number of total states in S. The reward function $r(i,a) \in R$ is the immediate reward while selecting Action a in State i. Then, the expected reward function J_α at i under policy μ, which is required to be maximized, is defined as the sum of the future rewards over an infinite horizon, as discounted by Parameter α:

$$J_\alpha(i,\mu) := \int_0^\infty e^{-\alpha t} \sum_{j \in S} p_{i,j}(t, \mu(i)) r(j, \mu(j)) dt. \tag{3}$$

According to Theorem 4.6 in in Guo [2009], there exists an optimal policy μ^* that makes J_α to be maximized. By the definition of the value function J_α, Guo [2009] offers a policy iteration algorithm that can be used to obtain the optimal policy which is described as Algorithm 1.

3. PROBLEM STATEMENT

A given pursuit-evasion case can be modeled as a continuous-time markov process. In this section, a simple discrete pursuit-evasion example will be introduced and then the details to construct the CTMP will be described.

3.1 Dynamics

Consider the environment as a discrete grid of 2 dimension plane, in which two robots try to track and evade from each other. For simplicity, the robots move simultaneously step by step. Each step of the movement is called an action. In this context, the robots are considered to have

Algorithm 1 Policy Iteration Algorithm

1: Pick an arbitrary μ. Let $k = 0$ and take $\mu_k = \mu$.
2: (Policy evaluation).
3: Obtain $R(\mu_k) = [r(1, \mu_k(1)), ..., r(|S|, \mu_k(|S|))]^T$
4: Obtain $J_k = [\alpha I - Q(\mu_k)]^{-1} R(\mu_k)$.
5: (Policy improvement). Obtain a policy $\mu_{k+1} = a_1, a_2, ..., a_{|S|}$ that provides $\frac{r(i,a_i)}{\alpha + q(i|i,a_i)} + \frac{1}{\alpha + q(i|i,a_i)} \sum_{j \neq i} q(j|i,a_i) > J_k(i)$, for $\forall i \in S$.
6: **if** $\mu_{k+1} = \mu_k$ **then**
7: stop
8: **else**
9: go to Step 2
10: **end if**

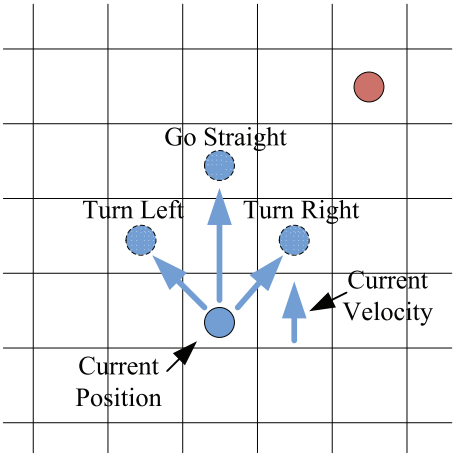


Fig. 1. The environment and the actions of robots

three available actions $a \in \{L, R, S\}$, namely "turn left", "turn right" and "go straight"(Fig. 1). What the present paper focuses on is how to find successive actions for the blue robot, which finally makes him arrive in the back region of the red robot. The environment is represented as $\Omega = \{(x,y)|x,y \in \mathbf{N}\}$. The state of the robot's dynamic consists of the position and the velocity information. As an example, the state vector of the blue robot is represented as $x_b = (x^b, y^b, v_x^b, v_y^b)$, and the entries of x_b denote the horizontal and vertical positions and velocities respectively. Assuming that $(x^b, y^b) \in \Omega$ and $(v_x^b, v_y^b) \in \{v_x^b + v_y^b = 1 \; and \; v_x^b, v_y^b = 0, 1\}$, the dynamic of the robots can be written as Algorithm 2:

3.2 Combat-States, Goal and Rewards

The pursuit-evasion process can be considered as a combating process. In the previous subsection, the dynamics of a single robot have been introduced, in which only the single side is considered, while the opponent is ignored. In order to depict the combating process, the Combat-States in relation to the dynamics of both robots are defined.

The Combat-State is represented as $x_c = [d, AA, ATA]$, where d is the Manhattan distance(It will be more suitable for the discrete environment in this paper), and ATA and AA are described in Fig. 2. Given dynamic states x_b and x_r of the blue and red robots, these elements in Combat-State can be obtained as shown in (4).

Algorithm 2 Robots Dynamics

if action ="turn left"$(a = L)$ **then**
  if $[v_x(k), v_y(k)] = [0, 1]$ **then**
    $x(k+1) = [x(k) - 1, y(k) + 1, -1, 0]$
  else if $[v_x(k), v_y(k)] = [-1, 0]$ **then**
    $x(k+1) = [x(k) - 1, y(k) - 1, 0, -1]$
  else if $[v_x(k), v_y(k)] = [0, -1]$ **then**
    $x(k+1) = [x(k) + 1, y(k) - 1, 1, 0]$
  else if $[v_x(k), v_y(k)] = [1, 0]$ **then**
    $x(k+1) = [x(k) + 1, y(k) + 1, 0, 1]$
  end if
else if action ="turn right"$(a = R)$ **then**
  if $[v_x(k), v_y(k)] = [0, 1]$ **then**
    $x(k+1) = [x(k) + 1, y(k) + 1, 1, 0]$
  else if $[v_x(k), v_y(k)] = [1, 0]$ **then**
    $x(k+1) = [x(k) + 1, y(k) - 1, 0, -1]$
  else if $[v_x(k), v_y(k)] = [0, -1]$ **then**
    $x(k+1) = [x(k) - 1, y(k) - 1, -1, 0]$
  else if $[v_x(k), v_y(k)] = [-1, 0]$ **then**
    $x(k+1) = [x(k) - 1, y(k) + 1, 0, 1]$
  end if
else if action ="go straight"$(a = S)$ **then**
  $x(k+1) = x(k) + [2v_x(k), 2v_y(k), 0, 0]$
end if

$$\begin{aligned} d &= |x^b - x^r| + |y^b - y^r| \\ ATA &= \pm arccos \frac{(v_x^b, v_y^b) \cdot (x^r - x^b, y^r - y^b)}{|(v_x^b, v_y^b)| \cdot |(x^r - x^b, y^r - y^b)|} \\ AA &= arccos \frac{(v_x^r, v_y^r) \cdot (x^r - x^b, y^r - y^b)}{|(v_x^r, v_y^r)| \cdot |(x^r - x^b, y^r - y^b)|} \end{aligned} \quad (4)$$

This group of equations above offers a description of the Combat-States as represented by the dynamic states of the two robots. The size of the problem is limited by integer N, where it provides that $|x^r - x^b| \leq N$ and $|y^r - y^b| \leq N$. It is obviously to see that AA is mainly depends on the red robot and ATA is the blue one. The range of d belongs to $[1, 2N]$, AA is set to $[0, \pi]$ and ATA is set to $[-\pi, \pi)$. Because if the range of ATA is $[0, \pi]$, it is impossible to distinguish which side of LOS for v_b has been located. From the direction of the LOS to v_b, if the direction is counter-clockwise then $ATA > 0$, otherwise, $ATA < 0$,which can help to calculate the next position with the selected action in the algorithm of the reward function(see.Alg.2). However in the calculation of the likelihood functions and transition rates,whether ATA is positive or negative is ignorable and the range of ATA will be set to $[0, \pi]$.

To achieve this goal, the reward functions are necessary for obtaining a solution in the mathematic model.The outcome in State i is represented as $g(i)$, which consists of two parts, namely the goal zone reward function $g_{pa}(i)$ and the scoring function $S(i)$ McGrew et al. [2010]. The outcome with the weight $\omega_g \in [0, 1]$ can be defined as:

$$g(i) = \omega_g g_{pa}(i) + (1 - \omega_g) S(i). \quad (5)$$

Firstly, we define a goal zone, and reward the states that make the red robot drop in the goal zone and punish the failing one. The goal zone of the blue robot is assumed to be the four blocks in front of the robot, see Fig.3. If the opponent is in the goal zone, one unit of rewards can be obtained which provides $(g_{pa}(i) = 1)$; otherwise zero

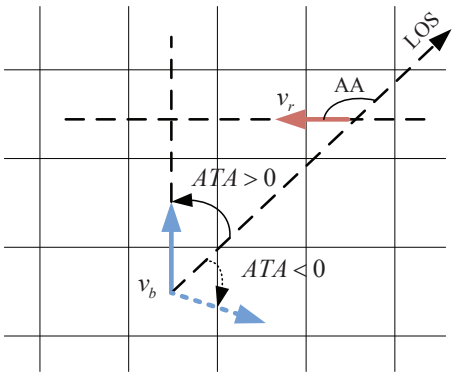


Fig. 2. The geometrical relationship of robots

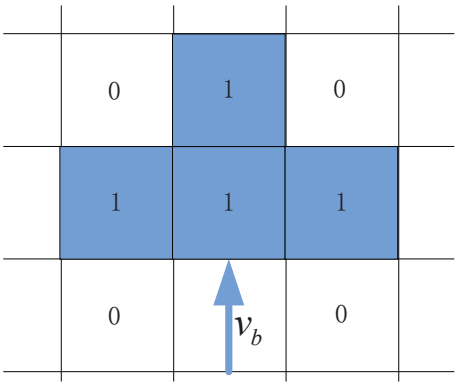


Fig. 3. The goal zone and g_{pa}

$(g_{pa}(i) = 0)$. Furthermore, it is also helpful for getting a better solution by defining a scoring function. The scoring function is an expert heuristic function, which reasonably captures the relative merits of every possible state in the adversarial game McGrew et al. [2010]. The scoring function can be evaluated by the elements of the Combat-State i, as shown in 6. Where $\bar{d}$ denotes the expected distance between the robots, which is set to $\bar{d} = 3$. The constant $\mathcal{K}$ is a factor to adjust the relative effect of the range and the angle, in this paper it is set to $\mathcal{K} = 1/\pi$.

$$S(i) = \frac{(1 - \frac{|ATA|}{\pi}) + (1 - \frac{|AA|}{\pi})}{2} \exp\left(\frac{-|d - \bar{d}|}{\pi \mathcal{K}}\right) \quad (6)$$

The reward functions $r(i, a)$ can be obtained through Algorithm 3.

4. CONSTRUCTION OF TRANSITION RATES MATRIX

As the Combat-States, actions, and rewards have been depicted, modeling the pursuit-evasion process into a CT-MDP with especial attentions on construction of the transition rates will be described in this section.

4.1 Combating Situations

Different from the MDP,in a pursuit-evasion process, it is obvious that the time of spent for transition will infect the combating situation to a great extent. Thus it is more reasonable if consideration is given to the connection between the transition probability and time. The transition probability $p(j|i, a)$ is replaced by $p(j|i, a; t)$, which mirrors

Algorithm 3 The reward function $r(i,a)$

Given the action a of the blue robot in the current Combat-State $i = [d, AA, ATA]$.
Assuming the red robot movies immobile, obtain $\bar{i} = [\bar{d}, \bar{AA}, \bar{ATA}]$.
$\bar{d} = d$ *and* $\bar{AA} = AA$
if $a = L$ **then**
  if $ATA > \frac{\pi}{2}$ **then**
    $\bar{ATA} > \frac{3\pi}{2} - ATA$
  else
    $\bar{ATA} = -\frac{\pi}{2} + ATA$
  end if
end if
if $a = R$ **then**
  if $ATA < -\frac{\pi}{2}$ **then**
    $\bar{ATA} = \frac{3\pi}{2} + ATA$
  else
    $\bar{ATA} = -\frac{\pi}{2} + ATA$
  end if
end if
if $a = S$ **then**
  $\bar{ATA} = ATA$
end if

the probability of transferring from State i to State j after time period t.

There are so many Combat-States that make the problem hard to be solved. Classifying these states may be not a bad choice. Similar with the cases in Virtanen et al. [2006, 2004], basing on the relative geometry of the states of the two robots, the concept of combating situation will be introduced here. The possible outcomes of combating situation in Step k are divided into four classes, which are neutral, defined by $\theta_k = \Theta_1$; advantage, defined by $\theta_k = \Theta_2$; disadvantage, defined by $\theta_k = \Theta_3$; and mutual disadvantage, defined by $\theta_k = \Theta_4$. Situation θ_k is a random variable that satisfies the following equation:

$$\sum_{l=1}^{4} p(\theta_k = \Theta_l) = 1.$$

The four above-mentioned combating situations are illustrated in Fig. 4. To be more specifical, in the neutral situation, the two robots are located far from each other, and in a not long future, the influence of the actions selected by both sides can be negative. There is no obvious intention that can be observed (Fig. 4(a)). Secondly, in the advantage situation, the blue robot occupies a dominated place, or in other words, the red robot is located in front of the blue one and the distance is at a middle level (Fig. 4(b)). Thirdly, the disadvantage situation is just to the contrary, in which the roles of both sides are exchanged (Fig. 4(c)). Finally, in the mutual disadvantage situation, the two robots move against to each other, and both of them are in disadvantage locations (Fig. 4(d)).

4.2 Likelihood Functions Depended on Situation

After the probability of the situation node is depicted, the transition probability $p(j|i,a;t)$ can be deduced by the Bayesian theorem, which provides a method for calculating the probability using the likelihood function of the situation:

$$\begin{aligned} &p(j|i,a;t) \\ &:= p(j(t)|i,a) = \sum_{k=1}^{4} p(\theta_i = \Theta_k|i) p(j|\theta_i = \Theta_k, a) \\ &= \sum_{k=1}^{4} \frac{p(\theta_i = \Theta_k) p(i|\theta_i = \Theta_k)}{\sum_{l=1}^{4} p(\theta_i = \Theta_l) p(i|\theta_i = \Theta_l)} p(j|\theta_i = \Theta_k, a) \end{aligned} \tag{7}$$

It is assumed that the elements of Combat-State are independent in the given situation, which can be represented as:

$$p(j|\theta_i) = p(d|\theta_i) p(ATA|\theta_i) p(AA|\theta_i). \tag{8}$$

Except for the neutral situation, items d, AA, and ATA are considered obeying the Gaussian distribution. The details of these distributions are listed in Table 1.

In the neutral situation Θ_1, there are no special preferences on the distribution of variables d, AA, and ATA. They could be any value as long as it belongs to the reasonable intervals. Thus, their likelihood functions are average distributions.

In the advantageous situation Θ_2, the blue robot is in domination, and a reasonable assumption is that the greater advantage the blue robot has, the smaller Variable d will be and to a greater extent the AA, and ATA will become close to the sight of line (LOS). It is assumed that these random variables obey 0-mean Gaussian distributions with different variances.

Contrastively, in the disadvantageous situation Θ_3, the variables also take the Gaussian form but the mean value ofAA, and ATA are both set to π.

Finally in the mutual disadvantageous situation Θ_4, the distribution of Variable d keeps the same and the mean value of AA, and ATA are set to π and 0 respectively.

Table 1. The conditional distributions on entries of Combat-State

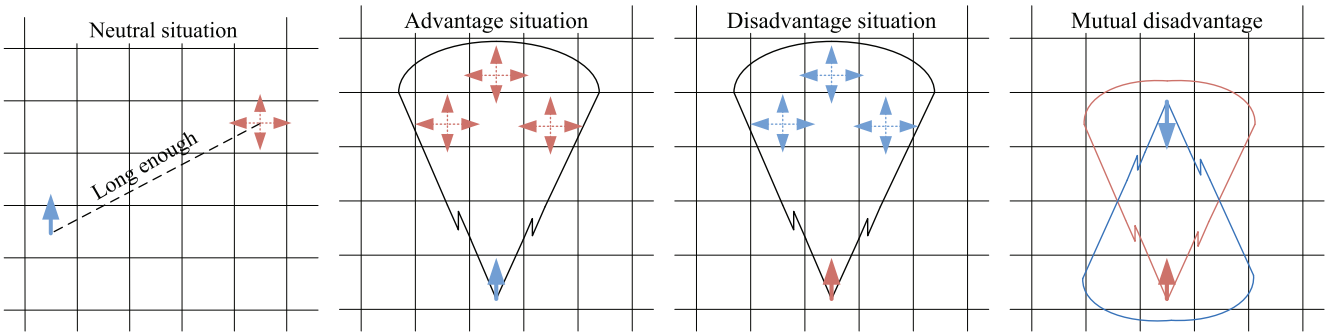
	$d \in [1, 2N]$	$AA \in [0, \pi]$	$ATA \in [0, \pi]$
Θ_1	$\frac{1}{2N-1}$	$\frac{1}{\pi}$	$\frac{1}{\pi}$
Θ_2	$\frac{1}{\sqrt{2\pi}\delta_d} e^{\frac{-d^2}{2\delta_d^2}}$	$\frac{1}{\sqrt{2\pi}\delta_a} e^{-\frac{AA^2}{2\delta_a^2}}$	$\frac{1}{\sqrt{2\pi}\delta_a} e^{\frac{-ATA^2}{2\delta_a^2}}$
Θ_3	$\frac{1}{\sqrt{2\pi}\delta_d} e^{\frac{-d^2}{2\delta_d^2}}$	$\frac{1}{\sqrt{2\pi}\delta_a} e^{-\frac{(\pi-AA)^2}{2\delta_a^2}}$	$\frac{1}{\sqrt{2\pi}\delta_a} e^{-\frac{(\pi-ATA)^2}{2\delta_a^2}}$
Θ_4	$\frac{1}{\sqrt{2\pi}\delta_d} e^{\frac{-d^2}{2\delta_d^2}}$	$\frac{1}{\sqrt{2\pi}\delta_a} e^{-\frac{(\pi-AA)^2}{2\delta_a^2}}$	$\frac{1}{\sqrt{2\pi}\delta_a} e^{\frac{-ATA^2}{2\delta_a^2}}$

4.3 Construction of Q

According to Stroock [2005], the solution of the Kolmogorov equation can be represented as follows:

$$P(t) = e^{tQ} = \sum_{m=0}^{\infty} \frac{t^m Q^m}{m!}, \quad t \in [0, \infty]. \tag{9}$$

Considering the extremely limited neighbor domain of $t = 0$, it can be obtained that the first order Taylor series expansion is $P(t) \approx I + tQ$, and then it can be inferred that $Q = \frac{dP(t)}{dt}$. On the other side, from (7) the probability can be constructed by the method depicted in the previous section. Thus, given i and j with $i \neq j$,



(a) Neutral situation (b) Advantage situation (c) Disadvantage situation (d) Mutual situation

Fig. 4. Four different situations in combating process

Probability $p(j|i,a;t)$ can be written as the weight sum of the condition probabilities

$$p(j|i,a;t) = \sum_{l=1}^{4} w_{i,l} p(j|\theta_i = \Theta_l, a) \tag{10}$$

in which

$$w_{i,l} = \frac{p(\theta_i = \Theta_l) p(i|\theta_i = \Theta_l)}{\sum_{\iota=1}^{4} p(\theta_i = \Theta_\iota) p(i|\theta_i = \Theta_\iota)}. \tag{11}$$

From (2) and (10), it can be obtained that

$$q(j|i,a) = \frac{dp(j|i,a;t)}{dt} = \frac{d}{dt} \sum_{l=1}^{4} w_{i,l} p(j|\theta_i = \Theta_l, a) \tag{12}$$

Equation (10) can be written into the first order Taylor series expansion, and then it can be found that the value of $q(j|i,a)$ is equal to the sum of the coefficients of monomial terms in the expansion of $p(j|\theta_i = \Theta_l, a)$. It is easy to note that the monomial term of $p(j|\theta_i = \Theta_1, a)$ is zero. $p(j|\theta_i = \Theta_3, a)$ is used as an example to show how to get Q. Based on table 1, it can be obtained as following:

$$\begin{aligned} p(j|\theta_i = \Theta_3, a) = & \frac{1}{\sqrt{8\pi^3}\delta_d \delta_a^2} \exp\left(\frac{(d + k_3^d(a)t)^2}{-2\delta_d^2}\right) \\ & \cdot \exp\left(\frac{(\pi - AA - k_3^{AA}(a)t)^2}{-2\delta_a^2}\right) \\ & \cdot \exp\left(\frac{(\pi - ATA - k_3^{ATA}(a)t)^2}{-2\delta_a^2}\right) \end{aligned} \tag{13}$$

If we define that

$\nabla_l(a) = [k_l^d(a)/\delta_d^2, k_l^{AA}(a)/\delta_a^2, k_l^{ATA}(a)/\delta_a^2], l = 2,3,4$ and $C_0 = \frac{1}{\sqrt{8\pi^3}\delta_d\delta_a^2}$ and $x_c(j) = [d, AA, ATA]$, then we have that:

$$\begin{aligned} p(j|\theta_i = \Theta_2, a) = & C_0(1 - \nabla_2(a) x_c(j)^T t) \\ & \cdot \exp\left(-\frac{d^2}{2\delta_d^2} - \frac{AA^2}{2\delta_a^2} - \frac{ATA^2}{2\delta_a^2}\right), \end{aligned} \tag{14}$$

$$\begin{aligned} & p(j|\theta_i = \Theta_3, a) = C_0(1 - \nabla_3(a)(x_c(j) - [0,\pi,\pi])^T t) \\ & \cdot \exp\left(-\frac{d^2}{2\delta_d^2} - \frac{(\pi - AA)^2}{2\delta_a^2} - \frac{(\pi - ATA)^2}{2\delta_a^2}\right), \end{aligned} \tag{15}$$

$$\begin{aligned} & p(j|\theta_i = \Theta_4, a) = C_0(1 - \nabla_4(a)(x_c(j) - [0,\pi,0])^T t) \\ & \cdot \exp\left(-\frac{d^2}{2\delta_d^2} - \frac{(\pi - AA)^2}{2\delta_a^2} - \frac{ATA^2}{2\delta_a^2}\right). \end{aligned} \tag{16}$$

And define

$$K_l(a) = [k_l^d(a), k_l^{AA}(a), k_l^{ATA}(a)], \quad for \quad l = 2,3,4,$$

as the change rates dependent on the maneuvering capabilities in d, AA, ATA. The transition rate can be represented as:

$$\begin{aligned} q(j|i,a) = & -C_0\{w_{i,2} \cdot \lambda_2 \cdot \nabla_2(a) x_c^T(j) \\ & + w_{i,3} \cdot \lambda_3 \cdot \nabla_3(a)(x_c(j) - [0,\pi,\pi])^T \\ & + w_{i,4} \cdot \lambda_4 \cdot \nabla_4(a)(x_c(j) - [0,\pi,0])^T\} \end{aligned} \tag{17}$$

Where

$$\lambda_2 = \exp\left(-\frac{d^2}{2\delta_d^2} - \frac{AA^2}{2\delta_a^2} - \frac{ATA^2}{2\delta_a^2}\right)$$

,

$$\lambda_3 = \exp\left(-\frac{d^2}{2\delta_d^2} - \frac{(\pi - AA)^2}{2\delta_a^2} - \frac{(\pi - ATA)^2}{2\delta_a^2}\right)$$

and

$$\lambda_4 = \exp\left(-\frac{d^2}{2\delta_d^2} - \frac{(\pi - AA)^2}{2\delta_a^2} - \frac{ATA^2}{2\delta_a^2}\right).$$

To provide the CTMDP a solution, it is needed to sustain the assumption of the conservativeness (1),i.e.

$$q(i|i,a) = \sum_{j \neq i} q(j|i,a). \tag{18}$$

5. NUMERICAL EXAMPLES

To verify the effectiveness of our proposed CTMDP-based method on the pursuit-evasion problem, some numerical experiments are recorded in this section. Here the size of the combating window is set to be $N = 5$, and the parameters selected in numerical examples are listed in table 2.

5.1 Keep-Straight Situation

In the first numerical example, the predetermined path of the red robot is linear. The position and direction are initialized to $x_r(0) = [5,5,-1,0]$. The x-y coordinates are [5,5], and the direction is left. For the blue robot, there are four different initial states, where $x_b(0)$ is set to [7,7,-1,0], [7,3, 1,0], [3,1, -1,0], and [3,9, 1,0].

The problem with the CTMDP is solved by the method given in the previous sections, and an efficient policy can thus be obtained to ensure the blue robot respond to the red ones actions (See Fig. 5). It is shown that the policy helps the blue robot catch up with the red one in a short path under the four different initial states respectively.

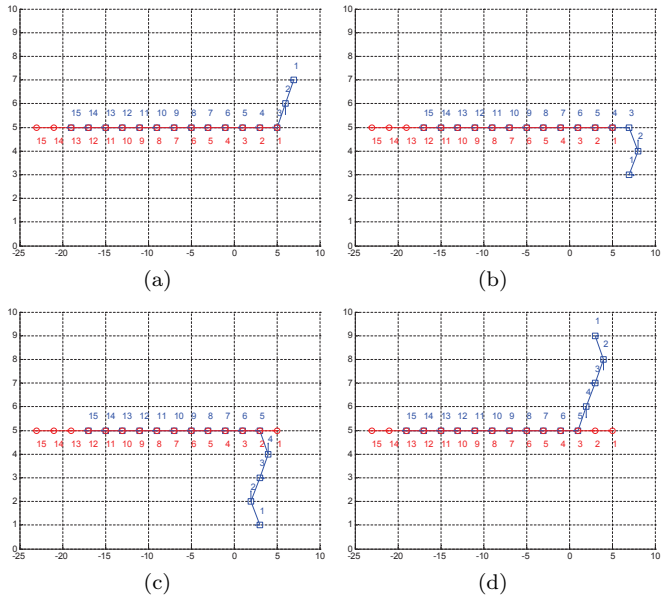


Fig. 5. The numerical results in keep-straight situation

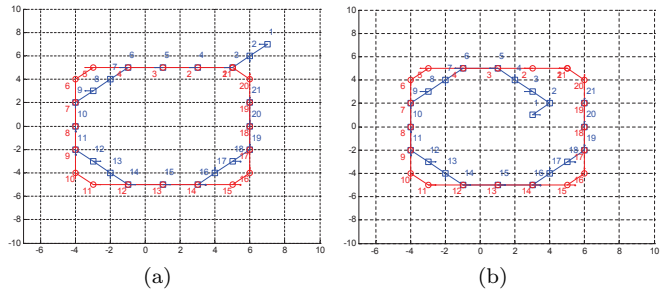


Fig. 6. The numerical results in keep-turning situation

5.2 Keep-Turning Situation

In this section, the red robot moves in a circle, as shown in Fig. 6. The initial state of the red one is also set to $x_r(0) = [5, 5, -1, 0]$. Two typical states for $x_b(0)$ are selected: one inside the circle and the other outside, which are [7,7,-1,0] and [3,1, 1,0], respectively. Fig. 6 shows that the blue robot is trying to track the red one as soon as possible, or in other words, trying to locate the red one in the goal zone of the blue one.

Table 2. The parameters used in numerical examples

Parameter	Value
$[w_{i,2}, w_{i,3}, w_{i,4}]$	[0.25,0.25,0.25]
$[w_g, d, \alpha]$	[0.5, 3, 0.8]
$[\delta_d, \delta_a]$	[3, $5\pi/18$]
$[K_2^T(a=1), K_2^T(a=2), K_2^T(a=3)]$	[-3, -3, -3.6 $-5\pi/18$, $-5\pi/18$, 0 $-5\pi/18$, $-5\pi/18$, 0]
$[K_3^T(a=1), K_3^T(a=2), K_3^T(a=3)]$	[-3, -3, -3.6 $5\pi/18$, $5\pi/18$, 0 $5\pi/18$, $5\pi/18$, 0]
$[K_4^T(a=1), K_4^T(a=2), K_4^T(a=3)]$	[-3, -3, -6 $5\pi/18$, $5\pi/18$, 0 $-5\pi/18$, $-5\pi/18$, 0]

6. CONCLUSION

In this paper, the pursuit-evasion process is modeled into a continuous time Markov decision process. Based on the situations, a Bayesian approach is used to describe the transition probability matrix, and an efficient method is presented to address how to construct transition rates in such a context.

In both numerical examples, the successive control decisions of the blue robot are made appropriately. The results indicate that this method provides a feasible solution. The future work would focus on extending it to a 3-D form. A potential difficulty in achieving this goal will be caused by the fact that the size of pursuit-evasion will certainly increase, which will be accompanied by the increase in the number of state variables and that of control actions. Consequently, the method for developing a policy in the CTMDP should be extended to a more complicated one.

REFERENCES

Bertsekas, D. and Tsitsiklis, J. (1996). *Neuro-Dynamic Programming.* Anthropological Field Studies. Athena Scientific.

Faiya, B., Carleton University. Dissertation. Engineering, E., and Computer (2012). *Learning in Pursuit-evasion Differential Games Using Reinforcement Fuzzy Learning.* Carleton University.

Guo, X.P. (2009). *Continuous-Time Markov Decision Processes.* Heidelberg: Springer.

Isler, V. and et al. (2005). Roadmap based pursuit-evasion and collision avoidance.

LaValle, S. (2006). *Planning Algorithms.* Cambridge University Press.

McGrew, J.S., How, J.P., Williams, B., and Roy, N. (2010). Air combat strategy using approximate dynamic programming. *AIAA Journal on Guidance, Control, and Dynamics*, 33(5), 1641–1654.

Russell, S. and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach.* Prentice Hall series in artificial intelligence. Prentice Hall.

Shedied, S.A. (2002). *Optimal Control for a Two Player Dynamic Pursuit Evasion Game; The Herding Problem.* Ph.D thesis, Virginia Polytechnique Institute and State University, Virginia, USA.

Stroock, D.W. (2005). *An Introduction to Markov Process.* Heidelberg: Springer.

Virtanen, K., Karelahti, J., Raivio, T., and Ccc, T. (2006). Modeling air combat by a moving horizon influence diagram game. *Journal of Guidance, Control, and Dynamics*, 29(5), 1080–1091.

Virtanen, K., Raivio, T., and Hamalainen, R.P. (2004). Modeling pilot's sequential maneuvering decisions by a multistage influence diagram. *Journal of Guidance, Control, and Dynamics*, 27, 665–677.