Abstract—This paper investigates an optimal search problem under the assumption that a target to be found appears randomly and stays for a fixed time interval. We first formulate the search problem as an optimization problem to maximize the probability of finding the target while reducing the control energy consumption, and provide a solution to this problem. In this paper, we moreover derive the expectation value of how many times the target appears without being found and a necessary and sufficient condition for the agent’s behavior to converge to a periodic trajectory. The effectiveness of the proposed method is demonstrated through a numerical simulation.

I. INTRODUCTION

Search theory studies how to deploy an agent in order to find a target within finite resources. This theory is motivated by several practical applications such as detection of lost objects, rescue operations and medical services. Early works on search theory was given by Koopman [1] and Stone [2] and a large amount of research works have been devoted to this theory in some research fields such as operations research, artificial intelligence and so on, which are summarized in the survey paper [3].

In recent years, the search problem receives a lot of attention in systems and control society. Mangel [5] formulated a so-called continuous search problem taking account of the agent behavior, which is known to be hard to solve. DasGupta et al. [6] presented an approximate solution to a continuous search problem for a stationary target by partitioning the search area into finite collections of regions. As another approach, Riehl et al. [7] proposed a graph-based model predictive search algorithm to minimize the expected time until the target detection. On the other hand, the search problem is investigated in the framework of pursuit-evasion games. Hespanha et al. [8] presented a search scheme guaranteeing that an evader is found in finite time under some assumptions. Chung et al. [9] formulated a search problem as a decision problem and presented some search strategies, where agents make decisions on whether or not the target is in the mission space. Frazzoli, Enright et al. [10], [11], [12] handles a search problem of stochastically-generated targets and presented a scheme minimizing the expected time from target appearance to detection. In addition, as a problem with a similar objective to the search problem, coverage control has been extensively studied [13], [14], [15]. The goal is to drive the sensors/agents to the position such that a given region is optimally covered by sensor networks.

In this paper, we present a novel search control scheme taking account of energy consumption. Indeed, reduction of energy consumption is an important problem in practice and the original definition of the search problem in [1], [2] includes this problem. We first formulate an optimal search problem maximizing the probability of finding a target while minimizing the energy consumption. We next give an approximate solution to the optimal search problem based on a coverage control scheme due to Li and Cassandras [14]. Then, a search control scheme is established based on receding horizon techniques. Moreover, we compute the expectation value of how many times the target appears without being found and give a necessary and sufficient condition for our control scheme to lead the agent to a periodic trajectory asymptotically.

The organization of this paper is as follows. In section III and IV, we propose a problem formulation of the optimal search control and provide its approximate solution. The expectation value of how many times the target appears without being found, and a necessary and sufficient condition for the agent’s state and control input to converge respectively to periodic trajectories are derived in section V. Next, we expand the proposed algorithm in section III and IV for multi-agent system in section VI. In section VII it is shown by simple simulations that the proposed method is effective.

II. PROBLEM SETTING

Let the search area \( E \subset \mathbb{R}^n \) \((n \in \{1,2,3\})\) be a bounded set. In this paper we mainly consider the planar case, i.e., \( n = 2 \). We assume that the target to be found appears randomly and stays for \( h[s] \) \((g \in \{1,2,\ldots\})\) and \( h[s] \) is a sampling period. The function \( \phi(z) \) is a density function which represents the probability that the target appears in \( z \in E \) and it satisfies \( \int_E \phi(z) dz = 1 \). As an example, \( \phi(z) \) can be expressed as

\[
\phi(z) = e^{-\gamma ||z-z_0||} \int_E e^{-\gamma ||z-z_0||} dz, \quad \gamma \geq 0
\]

where \( z_0 \) is the location which has the highest probability of target appearance. Given no prior knowledge of the distribution, we let \( \phi(z) \) be a uniform distribution.

Suppose that the agent equips a sensor and makes an observation of a target at prescribed time step \( t_k, k = 1,2,\ldots \) which is called observation time. For simplicity we assume that the observation time \( t_k \) is \( t_k = kh \) for positive integer \( h \). Here, \( y(t) \in \mathbb{R}^n \) is the agent’s position, and the position at time \( t_k \) is represented by \( y_k := y(t_k) \). Let the observation point set from time \( t_P \) to \( t_Q \) \((0 < t_P \leq t_Q)\) be denoted by

\[
Y_{P,Q} := \{y_k\}_{k=P,P+1,\ldots,Q}.
\]
As an example, $j$ increases. We represent a sensor model by a monotonically increasing in time, the agent should minimize the search level represented by a second order mass-spring-damper system, when the target appears, i.e., above $Y_j$ which represents the probability that the target is not found.

Moreover, \( \{a, b, c\} \) is described as

\[
Q(t) = 21 - e^{-\mu||z-y_k||^2}, \quad \mu > 0.
\]

Suppose that the target appears at time \( t_j \) (\( t_j < t_{j+1}, j \in \{0, 1, \ldots\} \)), and that the target exists in \( \mathcal{E} \) at \( t_k, k = j + 1, j + 2, \ldots, j + g \) (see Fig. 1). Then, the search level

\[
S(Y_{j+1:j+g}, \mathcal{E}) := \int_{\mathcal{E}} \phi(z) \prod_{k \in Y_{j+1:j+g}} p(||z-y_k||)dz,
\]

which represents the probability that the target is not found from \( t_{j+1} \) to \( t_{j+g} \) if the observation is performed at \( y_k \in Y_{j+1:j+g} \). However, the agent knows nothing about the time when the target appears, i.e., above \( j \).

Throughout this paper we assume that the agent motion is represented by a second order mass-spring-damper system, which is represented by

\[
\dot{x}(t) = Ax(t) + Bu(t), \quad x(t) = \begin{bmatrix} y(t) \\ \dot{y}(t) \end{bmatrix}
\]

where \( x(t) \in \mathbb{R}^{2n} \) is the state, \( y(t) \in \mathbb{R}^n \) is the velocity, \( u(t) \in \mathbb{R}^m \) is the control input, and the pair \((A, B)\) is controllable. Hereafter, the state and the velocity at time \( t_k \) are represented by \( x_k := x(t_k) \) and \( \dot{y}_k := \dot{y}(t_k) \), respectively.

**Remark 1:** The value of search level \( S \) does not depend on the order of the observation points, i.e.,

\[
Y_{P,Q} \in \text{sort}(Y_{P,Q}) \Rightarrow S(Y_{P,Q}, \mathcal{E}) = S(Y_{P,Q}, \mathcal{E}).
\]

### III. Optimal Search Control Problem

Because the agent does not know the target appearance time, the agent should minimize the search level \( S(Y_{j+1:j+g}, \mathcal{E}) \) for all \( j = 0, 1, \ldots \). This motivates the following finite-time optimal search control problem:

**Problem A:** Suppose that the current state \( x_k \) and the observation point sequence \( Y_{k:f} \) are given. Then for the system (4) and the prediction step \( f \in \{1, 2, \ldots\} \), find the optimal control input \( u(t) \in \mathcal{P}^m \) (\( \mathcal{P} \) denotes the set of all piecewise continuous functions), \( t \in [t_k, t_{k+f}] \) minimizing the cost function

\[
J_{k,k+f} := \sum_{i=k}^{k+f-1} J_u(u(t), t_i),
\]

\[
J_u(u(t), t_i) := \int_{t_i}^{t_{i+1}} u^T(t)Ru(t)dt, \quad R > 0.
\]

**Condition A:** Given a search area \( \mathcal{E} \) and a density function \( \phi(z) \), the observation point sequence \( Y_{k+1:k+f} \) locally minimizes

\[
\sum_{j=0}^{k+f-g} S(Y_{j+1:j+g}, \mathcal{E}).
\]

Condition A describes state constraints to minimize the sum of \( S(Y_{j+1:j+g}, \mathcal{E}) \), \( j = 0, 1, \ldots, k+f-g \) with respect to \( y_i, i = k+1, k+2, \ldots, k+f \). Thus Problem A is an optimal control problem to minimize the control energy consumption (6) under the state constraints.

### IV. Solution

**A. Reducing into the Discrete Optimization Problem**

In this section we first show that Problem A is reduced to a discrete optimization problem with respect to the observation point sequence \( Y_{k+1:k+f} \). For fixed \( x_i \) and \( x_{i+1} \) the optimal input \( u^*(t) \) to minimize (7) and the corresponding state trajectory \( x^*(t) \), \( t \in [t_i, t_{i+1}] \), and the resulting cost \( J^*_u(u^*(t), t_i) \) are given by

\[
\begin{align*}
Z(t) := & \begin{bmatrix} Z_{11}(t) & Z_{12}(t) \\ Z_{21}(t) & Z_{22}(t) \end{bmatrix}, \\
Z_{11}(t) := & -R^{-1}B^Te^{A(t_{i+1}-t)}W(h)^{-1}e^{Ah}, \\
Z_{12}(t) := & -R^{-1}B^Te^{A(t_{i+1}-t)}W(h)^{-1}, \\
Z_{21}(t) := & e^{A(t_{i+1}-t)}W(h)^{-1}e^{Ah}, \\
Z_{22}(t) := & e^{A(t_{i+1}-t)}(I - W(t_{i+1}-t)W(h)^{-1}), \\
M := & [e^{Ah}I] T W(h)^{-1} [e^{Ah}I], \\
W(t) := & \int_0^t e^{A(t-t')} B^TRB e^{A(t-t')} dt',
\end{align*}
\]

respectively (see e.g., [16]). Thus Problem A is reduced into the discrete optimization problem with the waypoint \( x_i, i = k+1, k+2, \ldots, k+f \). Here, \( Y \) and \( \dot{Y} \) denote the vertical concatenation of vectors \( \begin{bmatrix} y_{k+1}^T & y_{k+2}^T & \cdots & y_{k+f}^T \end{bmatrix}^T \).
and \([\mathbf{y}_{k+f}^T, \mathbf{y}_{k+2}^T, \ldots, \mathbf{y}_{kf}^T]^T\), respectively. Equation (6) can be rewritten as

\[
J_{k: k+f} = \sum_{i=k}^{k+f-1} J_{k} (\mathbf{u}(t), t_i)
\]

\[
= \sum_{i=k}^{k+f-1} \left[ x_i \right]^T M \left[ y_i \right] \left( \begin{array}{c}
\vdots \\
(10)
\end{array} \right)
\]

where \(H_1, H_2, H_3\) are matrices of dimensions \(n(f+2) \times n(f+2), n(f+2) \times n\) and \(n \times n\), respectively. Hence the optimal \(Y^*\) minimizing (11) and the minimal control energy consumption \(J_{k: k+f}^*\) are given by

\[
\dot{Y}^* = \left[ \begin{array}{c}
\dot{y}_{k+1}^* \\
\dot{y}_{k+2}^* \\
\vdots \\
\dot{y}_{kf}^*
\end{array} \right] = -H_3^{-1} H_2^T \left[ \begin{array}{c}
y_k^* \\
y_{k+1}^* \\
\vdots \\
y_{k+f}^*
\end{array} \right] 
\]

and

\[
J_{k: k+f}^* = \left[ \begin{array}{c}
y_k \\
y_k \\
\vdots \\
y_k
\end{array} \right]^T \left( \begin{array}{c}
H_1 - H_2 H_3^{-T} H_2^T
\end{array} \right) \left[ \begin{array}{c}
y_k \\
y_k \\
\vdots \\
y_k
\end{array} \right],
\]

respectively. Furthermore, from (9) and (12), the optimal control input \(u^*(t)\) and state \(x^*(t), t \in [t_k, t_{k+f}]\) are also given in an explicit form including the initial state and the observation point sequence as parameters (see Fig. 2).

In summary, the original problem (Problem A) is rewritten as the following optimization problem with respect to the observation point sequence \(Y_{k+1:k+f}\):

**Problem A':** Under Condition A, find the observation point sequence \(Y_{k+1:k+f}\) minimizing (13).

**B. Approximate Solution and Search Control Algorithm**

Problem A' is not easy to solve strictly. For this reason, we give a relaxed problem based on the following lemma.

**Lemma 1:** The following condition (14) is a sufficient condition for Condition A.

\[
Y_{1:g} \in \text{sort}(Y_{1:g}^\dagger), \quad y_k = y_{k-g}, \quad k = 1+g, 2+g, \ldots
\]

where \(Y_{1:g}^\dagger\) locally minimizes \(S(Y_{1:g}, \mathcal{E})\).

**Proof:** From the first condition of (14) and (5), \(S(Y_{1:g}, \mathcal{E}) = S(Y_{1:g}^\dagger, \mathcal{E})\). Moreover, the second equation of the condition (14), (5), \(Y_{1:g} = \{y_1\} \cup \mathcal{Y}_{2:g}\) and \(Y_{2:1+g} = \mathcal{Y}_{2:g} \cup \{y_{1+g}\}\) lead to \(S(Y_{1:g}, \mathcal{E}) = S(Y_{2:1+g}, \mathcal{E})\). By iterating above, we obtain \(S(Y_{j+1:j+g}, \mathcal{E}) = S(Y_{1:g}, \mathcal{E}), j = 0, 1, \ldots\). Therefore, if the condition (14) is satisfied,

\[
S(Y_{j+1:j+g}, \mathcal{E}) = S(Y_{1:g}, \mathcal{E}), \quad j = 0, 1, \ldots
\]

Since (8) is the sum of \(S(Y_{j+1:j+g}, \mathcal{E})\) and each term is minimized, Condition A is satisfied. This completes the proof.

Lemma 1 says that \(\mathcal{Y}_{\text{aff}} \subseteq \mathcal{Y}_{\text{eig}}\) where \(\mathcal{Y}_{\text{eig}}\) and \(\mathcal{Y}_{\text{aff}}\) express respectively the set of the observation point sequences satisfying Condition A and (14). In this sense, the observation point sequence obtained by solving Problem A subject to the condition (14) instead of Condition A is a suboptimal solution to Problem A'.

**Problem A':** Under the condition (14), find the observation point sequence \(Y_{k+1:k+f}\) minimizing (13).

It is possible to compute \(Y_{1:g}^\dagger\) efficiently by using a scheme of [14]. Problem A' can be solved by sorting the elements of \(Y_{1:g}^\dagger\) to minimize (13). However, the maximal computational complexity needed to solve it using enumerative method is given by \(O(g^n)\), and grows exponentially with respect to \(g\). Hence, we use so-called ant colony optimization (ACO) techniques [17] to gain its approximate solution, where the maximal computational complexity is \(O(g^n)\).

Table I shows the optimal search control algorithm. A receding horizon policy is used for control of the agent, that is, at time \(t_k\) the optimal control \(u(t), t \in [t_k, t_{k+f}]\) is computed, and \(u(t)\) is applied only for \(t \in [t_k, t_{k+1}]\).

**V. MAIN RESULT**

**A. Probabilistic Analysis of Target Detection**

We first give the following theorem concerning the time required to find the target.

**Theorem 1:** Let \(F(\beta)\) and \(E[\beta]\) be respectively the probability of detecting the target until the target appears \(\beta\) times
and the expectation value of the number of target appearance $\beta$ by the target is detected. Then $F(\beta)$ and $\mathbb{E}[\beta]$ are given by $F(\beta) = 1 - \alpha_\beta^g$ and $\mathbb{E}[\beta] = 1/(1 - \alpha_\beta^g)$.

**Proof:** From (15), $S(y_{j+1}, \mathcal{E}) = S(y_{j+1}, \mathcal{E}) =: \alpha_g$ $(0 \leq \alpha_g < 1)$, $j = 0, 1, \ldots$ is satisfied. This means that the probability for missed the target detection is the local minimum $\alpha_g$ whenever the target appears. Since the probability of finding the target for the first time when the target appears $b$ times is given by $\alpha_g^{b-1}(1 - \alpha_g)$, we obtain

$$F(\beta) = \sum_{b=1}^{\beta} \alpha_g^{b-1}(1 - \alpha_g) = 1 - \alpha_\beta^g.$$

$$\mathbb{E}[\beta] = \lim_{\beta \to \infty} \sum_{b=1}^{\beta} b\alpha_g^{b-1}(1 - \alpha_g) = \lim_{\beta \to \infty} \frac{1 - \alpha_\beta^g}{1 - \alpha_g} = \frac{1}{1 - \alpha_g}.$$

**\(B. Convergence to Periodic Trajectory\)**

We next analyze the behavior of the agent controlled by our present algorithm in Section III and IV. Indeed, it is desirable for the agent to take ordered motion from the viewpoint of dependability and reliability. In this paper, we thus provide a necessary and sufficient condition for the agent to converge to a periodic trajectory with a period $T = gh[s]$.

Let us now define the matrix $G$ by

$$G := \begin{bmatrix} I_n \\ O_n \\ \vdots \\ O_n \end{bmatrix}^T \begin{bmatrix} O_n \\ \vdots \\ O_n \end{bmatrix} - \begin{bmatrix} I_n \\ O_n \\ \vdots \\ O_n \end{bmatrix}^T H_3^{-1} H_2^T \begin{bmatrix} O_n \\ \vdots \\ O_n \end{bmatrix}$$

(16)

where $I_n$ and $O_n$ are an identity matrix and a zero matrix of dimensions $n \times n$, respectively. Roughly speaking, this matrix represents the effect of $y_k$ on $y_{k+1}$.

**Theorem 2:** Suppose that our control algorithm is applied to the agent with dynamics (4). Then the trajectories of the agent’s state and control input converge respectively to periodic ones with a period $T = gh[s]$ as $t \to \infty$ if and only if $\max_i |\lambda_i| < 1$, where $\lambda_i, i = 1, 2, \ldots, n$ are the eigenvalues of $G$.

**Proof:** Let the error vector $e[k]$ be defined by $e[k] := \hat{y}(t_k + T) - \hat{y}(t_k) = y_{k_{kg}} - y_k$, where

$$y_{k+1} = \begin{bmatrix} I_n \\ O_n \\ \vdots \\ O_n \end{bmatrix}^T \begin{bmatrix} O_n \\ \vdots \\ O_n \end{bmatrix} - \begin{bmatrix} I_n \\ O_n \\ \vdots \\ O_n \end{bmatrix}^T H_3^{-1} H_2^T \begin{bmatrix} O_n \\ \vdots \\ O_n \end{bmatrix} y_k.$$

Since $y_{k_{kg}} = y_k, k = 1, 2, \ldots$ from the second equation of (14), the error vector $e[k+1]$ is rewritten as

$$e[k+1] = y_{k_{kg}} - y_k.$$

Therefore, the discrete-time linear system $e[k+1] = Ge[k]$ is asymptotically stable, i.e., $\lim_{k \to \infty} e[k] = \lim_{k \to \infty} (y_{k_{kg}} - y_k) = 0$ if and only if $\max_i |\lambda_i| < 1$. Additionally, from the second equation of (14), $\lim_{k \to \infty} (x_{k_{kg}} - x_k) = 0$. Since $u(t)$ and $x(t)$ are given in an explicit form including the waypoints $x_k$ as parameters from (9), this proof is complete.

**Remark 2:** Note that $\max_i |\lambda_i|$ is a measure of the speed of convergence to a periodic trajectory.

**C. Discussion**

The above results give us an intuitive insight into the agent’s optimal strategy.

In the case of $g = 1$, which corresponds to the case where the target motion is quite fast, $y_k = y_k^i = \arg\min_j y_j S(\{y_j\}, \mathcal{E})$, $k = 1, 2, \ldots$ holds. This implies that the optimal trajectory which the agent can take is to stay at the location with the highest probability for the target detection. For example, in whack-a-mole, we should wait for a mole at the center, not with the dispersive focus of attention.

In the case of $g = 2, 3, \ldots$, the agent position trajectory converges to a periodic motion with a period $gh$. This means that the agent should go around the region widely as the target motion becomes slow. For example, in beetle hunting, we should move around and search the tree where a beetle will exist, not keep still. Moreover, even if the tree has been looked, we may be able to find beetles later.

**VI. COOPERATIVE SEARCH STRATEGY OF MULTI-AGENT SYSTEM**

When the target moves fast, that is, $g$ becomes small, single agent cannot search sufficiently and needs to move fast if the search area is large. Hence we consider a cooperative search problem of multi-agent system. In multiple-agent case, a new search method needs to be developed. In this section we present two cooperative search strategies: Voronoi
Voronoi based Cooperative Search Algorithm

1: \( k \leftarrow 0 \)
2: while 1
3: \( \text{if } \text{mod}(k, g) = 0 \)
4: Compute \( V(y_k^A) \) and \( C_{V_i}(y_k^A) \)
5: end if
6: Compute \( u(k^1(t), t \in [t_k, t_{k+1}]) \) by solving Problem B
7: Input \( u(k^1(t), t \in [t_k, t_{k+1}]) \)
8: \( k \leftarrow k + 1 \)
9: end while

A. Voronoi based Cooperative Search Algorithm

Each agent optimally search the responsible area for the target while sharing the search area \( E \) optimally.

Given the agent positions \( y_k^A := \{y_k^{(1)}, y_k^{(2)}, \ldots, y_k^{(n_a)}\} \), the Voronoi partition of \( E \) is \( V(y_k^A) := \{V_1(y_k^A), V_2(y_k^A), \ldots, V_{n_a}(y_k^A)\} \). For the system (4) and the prediction step \( y ▷ \leftarrow y_{k+1} \), the Voronoi partition of \( E \) is optimally shared. Suppose a team of \( n_a \) agents tries to find the target, and the subscript \( (l) \) means the variables of \( l \)-th agent.

Let the agent set be denoted by \( A := \{1, 2, \ldots, n_a\} \).

B. Distributed Cooperative Search Algorithm

Next, we present a distributed cooperative search strategy to try to locally minimize \( S(\bigcup_{l=1}^{n_a} y(l)_{k+1}) \) as \( k \rightarrow \infty \) with communication range \( D > 0 \). In this subsection, each agent is assumed to receive the observation point information from its neighbors.

The neighbor set of agent \( l \) at time \( t_k \) is defined as
\[
N(l) := \left\{ t \in A \left| ||y(l)^k - y(l)^t|| \leq D \right. \right\}.
\]

Let the \( l \)-th agent’s observation point information at time \( t_k \) be denoted by \( B(l)^k \), and it is updated at time \( t_k \) as
\[
B(l)^k = B(l)^{k-1} \cup \bigcup_{t \in N(l)} y(l)^t, j \in I_k
\]

where \( B(l)^0 = \emptyset \) and \( I_k \) is defined by \( I_k = \{k\} \) or \( I_k = \{1, 2, \ldots, k + g - 1\} \). The former means each agent receives the visual information, i.e., its neighbors’ current position, and the latter means the agents communicate the own past, current and future observation points from time \( t_1 \) to \( t_{k+1} \) with each other. Additionally, let the observation point information from time \( t_P \) to \( t_Q \) be denoted by
\[
B(l)^{P,Q} := \{y(l)^t, j \in P, t_{P+1}, \ldots, Q\}.
\]

The brief description of Voronoi based Cooperative Search Algorithm Problem B can be solved as well as Problem A.
**Problem C:** Suppose that the current state $x_k^{(i)}$ and the observation point set $Y_{k+1,k+g}$ are given. Then for the system (4) and the prediction step $f$, find an optimal control input $u(t) \in PC^m$, $t \in [t_k,t_{k+1}]$, minimizing the cost $J_{k,k+g}$, s.t., $Y_{k+1,k+g} \in \text{sort}(Y_{k+1,k+g})$ and $y_{k+g}^{(i)} = y_{k+g}^{(i)}$, $i = k + g, k + 1 + g, \ldots, k + f$.

3) Compute $u(t), t \in [t_k,t_{k+1}]$ and apply $u(t)$ only for $t \in [t_k,t_{k+1}]$.

Table III shows the Distributed Cooperative Search Algorithm, where $y_{int\_random}(g)$ is a function to randomly locate the initial observation points.

**Remark 3:** For $Z_k = [k]$, the proposed algorithm locally minimizes $S \left( \bigcup_{j=1}^{g} Y_{j_{k+1},k+g}^{(i)} \right)$ as $k \to \infty$ if there exists $k'$ satisfying $N_{k'}^{(i)} = \mathcal{A} \forall l \in \mathcal{A}, k = k', k'+1, \ldots$. In a similar way, for $Z_k = \{1, 2, \ldots, k+g-1\}$, it is true if there exists $k'$ satisfying $\bigcup_{j=k-g+1}^{k} N_{j}^{(i)} = \mathcal{A} \forall l \in \mathcal{A}, k = k', k'+1, \ldots$.

**VII. Simulations**

In this section, we demonstrate the effectiveness of the three proposed algorithm through numerical simulation.

**Example 1 (single agent):** The search area is $\mathcal{E} = [0, 4] \times [0, 5]$, and $\phi(z)$ is a uniform distribution ($\phi(z) = 1/20$). As a sensor model, we use the function (2) with $\mu = 1$. The dynamics of the agent (4) is given by

$$
\dot{x}(t) = \begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & -a & 0 \\
0 & 0 & 0 & -a
\end{bmatrix} x(t) + \begin{bmatrix}
0 \\
0 \\
1 \\
0
\end{bmatrix} u(t)
$$

where $a = 1$ is a viscosity coefficient. The initial state is $x_0 = [0.5, 0.5, 0, 0]^T$, the observation interval is $h = 1[s]$, the prediction step is $f = 5$, $g = 4$, and the cost parameter is $R = \text{diag}(1, 1)$. In this case, $\lambda_i = -0.2534$ (multiplicity 2), and the agent’s behavior converges to a periodic trajectory (period $T = 4[s]$) from Theorem 2.

Figs 3 and 4 show the comparison between the Optimal Search Control Algorithm (Table 1) and the random walk based method (choose $y_{k+1} \in \mathcal{E}$ randomly, and compute a control input from (9) and (12)). The green curves, red curves and the symbol $\times$ respectively describe the observation points, agent trajectories and the initial positions. We see from Fig. 3 that the position trajectory converges to a periodic one. Figs 5 and 6 describe the search level $S(\mathcal{Y}_{j_{k+1},j_{k+g}}, \mathcal{E})$ and control energy for both strategies. We see from these figures that the present algorithm achieves a better search performance within less energy consumption than the random walk based method. Furthermore, as shown in Fig. 5, $\alpha_g \approx 0.465$ in this example and we get $\mathbb{E}[\beta] \approx 1.870[\text{times}]$ from Theorem 1.

**Example 2 (single agent):** We next show another example of single agent. Let $\alpha = 5, \mu = 2, g = 18, f = 20$ and $\phi(z)$ be given by (1), where $\gamma = 0.3$ and $z_0 = [2.5, 3]^T$. In this case, $\lambda_i = -0.1182$ (multiplicity 2).

Figs 7 - 10 show the simulation result as well as Example 1. In Figs 7 and 8, the shade represents the high probability area of target appearance. We see from Fig. 7 that the agent searches intensively the locations which have the high probability for the target to appear, and the position trajectory converges to a periodic one with a period $T = 18[s]$. Furthermore, in Figs 9 and 10 the present algorithm is better as well as Example 1. Additionally, $\alpha_g \approx 0.1820$ and $\mathbb{E}[\beta] \approx 1.223[\text{times}]$. 

**Fig. 3** Proposed method ($g = 4$)

**Fig. 4** Random walk based method

**Fig. 5** Search level $S(\mathcal{Y}_{j_{k+1},j_{k+g}}, \mathcal{E})$

**Fig. 6** Control energy consumption $J_{0,k}$
Example 3 (Voronoi based Cooperative Search): We simulate a cooperative search with three agents, where $\mu = 1.5$, $f = 10$, $x_0^{(1)} = [0.5, 0.5, 0, 0]^T$, $x_0^{(2)} = [2.5, 1, 0, 0]^T$ and $x_0^{(3)} = [1, 3.5, 0, 0]^T$.

Fig. 11 shows the position trajectories $y^{(i)}(t)$, $t \in (t_{k-g}, t_k]$, the observation points $\mathcal{Y}_{k-g+1:k}$ and the responsible area $\mathcal{V}_l(y_k^{(i)})$ by using the Voronoi based Cooperative Search Algorithm (Table II). The square describes the agent position. As shown in Fig. 11(f), both dividing the search area and searching in its responsible area are achieved appropriately.

Example 4 (Distributed Cooperative Search): This example shows a simulation result of the Distributed Cooperative Search Algorithm (Table III) with $n_a = 3$, $\mu = 2$, $f = 10$, $g = 6$, $D = 3$, $x_0^{(1)} = [0.5, 0.5, 0, 0]^T$, $x_0^{(2)} = [4.5, 1, 0, 0]^T$ and $x_0^{(3)} = [1, 3.5, 0, 0]^T$.

Figs 12 and 13 show respectively the results for $\mathcal{I}_k = \{k\}$ and $\mathcal{I}_k = \{1, 2, \ldots, k+g-1\}$. We see from these figures that the latter scheme achieves a better search performance than the former.

VIII. Conclusions

In this paper, we have presented novel search algorithms under the situation that the agent dynamics is represented by a second order mass-spring damper system and a target to be found appears randomly and stays in a fixed time interval. We first have formulated an optimal search control problem and given its approximate solution. Then, we have presented the expectation value of how many times the target appears without being found. Moreover, we have clarified a necessary and sufficient condition for the agent to converge to a periodic trajectory. Next, we have presented two search
algorithms for multi-agent case. Finally, the effectiveness of our algorithms has been demonstrated through numerical simulation.

REFERENCES