This is the published version (version of record) of:

Pathirana, Pubudu, Black, Timothy and Nahavandi, Saeid 2005, Terminal approach
navigation for autonomous robots in a sensory network, in 2005 3rd IEEE International

Available from Deakin Research Online:

http://hdl.handle.net/10536/DRO/DU:30005651

Copyright: ©2005 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.
Terminal Approach Navigation for Autonomous Robots in a Sensory Network

Pubudu N. Pathirana  
School of Engineering and Technology  
Deakin University  
Victoria, Australia 3217  
Email: pubudu@deakin.edu.au

Timothy J. Black  
School of Engineering and Technology  
Deakin University  
Victoria, Australia 3217  
Email: tblack@deakin.edu.au

Saeid Nahavandi  
School of Engineering Technology  
Deakin University  
Victoria, Australia 3217

Abstract—In this paper we present a technique based on precision guidance approach for the sensor delivery and reception problem between two mobile robots. A slave robot is employed to collect sensors and stack them on a tray carried by the mobile master robot. Here we define the terminal attitude of the slave robot with respect to the master and present a LQR control approach to solving the problem of achieving a desired terminal approach angle necessary for the appropriate sensor delivery. The approach criteria is defined in terms of both minimizing the miss distance and controlling the slave robot's body attitude with respect to the master robot at the terminal point.

I. INTRODUCTION

Recent years have witnessed a burgeoning amount of research and commercial activities in the area of robotics in sensor networks, fueled by technological advances and potential applications[2], [4], [3]. Technological advances have fostered the emergence of a range of small, low-power devices that integrate microsensing, actuation, on-board processing, and wireless communications capabilities[4], [3]. Such devices can be deployed in large numbers and can form sensor networks to coordinate in performing higher-level tasks, enabling them to measure aspects of the physical environment in unprecedented detail. Furthermore, applications of sensor networks span many potential areas such as environmental monitoring, condition-based maintenance, military surveillance and bushfire alerts using arbitrarily spread smart sensors. These can be used in smart spaces, transportation, precision agriculture, inventory tracking, factory instrumentation, locating inmates of an retirement village after a panic alarm is activated.

Sensor networks are frequently characterized by extremely limited end-node power, memory and CPU capability. In addition, they are envisioned to exist at tremendous scale, with possibly thousands of nodes per network. Due to limited on-board resources of sensors, rather than communicating sensor information over long distances, communication within these nodes is often scheduled to conserve power, and sets of nodes are frequently names (or addressed) only in aggregate. The work presented in this paper concentrates on employed two robots to collect sensors in the data collecting sensory network. These sensors were deployed indiscriminately and the task of the two robots to collect the sensors. The localization of the sensors were addressed in [8], and this was implemented on a mobile robot (robot 1 in figure III.1) travelling in the sensory network. In this work, we employ another associate robot (robot 2 in figure III.1) collecting the sensors once the location has been established and placing on the sensors on robot 1. The underlying problem of sensor collecting robot approaching the main robot has been addressed using precision guidance ideas used in controlling autonomous bodies.

For these networks, a collection of powerful autonomous agents (i.e. flying devices/aircrafts) can roam around the network and collect data. Devices such as Robomote[14] are already in production and be used as mobile-robot. Moreover, these mobile robots can also be used to dynamically reprogram or reconfigure the sensors. Due to the large number of sensors involved, sensor deployment and operation spans several possibilities. In many applications, sensors can be dropped from an aircraft in the field or scattered by a slingshot. We assume that small, low-power sensors will not be equipped with GPS receivers. Therefore, sensors will not know their location during deployment. Applications of disconnected sensor networks already exist in practice [13], [7], [5].

One such project is the DataMules project[13]. It proposes a scheme wherein a Mule periodically visits sensor devices and collects information from these devices, in effect providing a message store-and-forward service, enabling low-power sensor nodes to conserve power. Another project is the Sami Network[7]. Sami people are reindeer herders in Sweden who keep relocating their base. The relocation is controlled by an yearly cycle which depends on the natural behaviour of reindeer. The Sami communities don't have wired or wireless communication infrastructure. In Zebranet[5], wireless sensor nodes attached to animals collect location data and opportunistically report their histories when they come within radio range of base stations.

Not surprisingly, most of the research on localization systems for ad hoc networks have been motivated by the

552
II. AUTONOMOUS ROBOTS

Much of the emphasis has been in designing location systems that allow devices to compute their own location, allowing the system to be scalable and responsive. However, requirements for localization for these systems are different from the application scenarios that we have discussed.

In this application the two robots we use are, the PowerBot(robot 1) and Pioneer 3-AT(robot 2)\(^1\). The PowerBot is an extremely maneuverable, high payload, high speed robotic mobile platform. With a high degree of intelligence, and the ability to carry a payload up to 100kg, the PowerBot is a smart robotic platform designed for research use.

It has the ability to perform, for example, material handling, delivery, environment mapping and navigation. This allow the robot to be used for applications like monitoring or patrolling areas looking for intruders or disturbances, reconnaissance tasks, vision and object recognition and colour/object tracking.

With an object orientated dynamic programming environment, ARIA, allows complex programmable behaviors to be achieved, this opens up the possibilities for cooperation operations between robots.

The powerful PowerBot is constructed of an 85cm x 62.5cm x 43cm aluminum body over a steel frame. The two motors contain 500-tick encoders with rear casters and inflatable 25cm diameter tires, which are able to handle indoor, outdoor and in between environments. This configuration provides it with a differential drive system, which by only moving the wheels on one side, provides a turning circle radius of 54cm.

The ability to climb at a 15% grade and sills of 3cm provides it to perform in many indoor environments. On flat surfaces, it can reach speeds of 6kph, and at a slower speed, on flat terrains, it can carry payloads greater than 100 kg. This allows an operation time of approximately 2.5 hours on two fully charged 12V lead-acid batteries.

PowerBot has a wide array of onboard sensors including 28 sonar sensors, front and rear bumpers, compasses and tilt-position sensors and a SICK laser range finding scanner used for navigating and obstacle avoidance. As well as a stereo range finding camera and a pan and tilt CCD camera for vision processing.

With an integrated Hitachi H8S-based microcontroller for custom I/O applications, sonar boards and other electronics as well as and an integrated PC, communications include a wireless Ethernet connection that allows the robot to communicate with a host computer or robot-to-robot communication. With the addition of a 6 DOF (Degree of freedom) 2 kg payload manipulator, the PowerBot is a powerful processing unit capable of high end intelligence and autonomous behavior.

---

\(^1\)www.robots.activmedia.com

---

The Pioneer P3-AT is constructed from a 50cm x 49cm x 26cm aluminum body. Having an embedded computer on-board, allows the robot to perform onboard vision processing, Ethernet-based communications, DGPS(Differential Global Positioning Systems), and other autonomous functions.

The four motors (one on each wheel) use 66:1 gear ratios and contain 100-tick encoders. The skid-steer platform with 21.5cm diameter wheels are specifically designed for outdoor use in rough terrains. The skid-steer system allows rotation in one place by moving both wheels. This allows direct turning on a single spot and is ideal for a highly versatile all-terrain robotic platform.

Having the ability to climb at a 45% grade and sills of 9cm allows it to traverse any outdoor environment with ease. On smooth terrain, the P3-AT can move at speeds of 0.7\(\text{m s}^{-1}\) and carry payloads up to 30 kg at slower speeds.

Using a smaller battery system than the PowerBot, the P3-AT base can run 3-6 hours on three fully charged batteries. This allows outdoor use experimentation to be lengthy.

As in the PowerBot, the P3-AT’s includes a Hitachi H8S-based microcontroller for user I/O operations. It also has the ability to communicate with a PC client and other robots in the same way as the PowerBot.

III. AUTONOMOUS ROBOTS KINEMATIC MODEL

The work presented here uses the terminal attitude guidance problem[11] where the control objective is to minimize the target interceptor miss distances in addition
to satisfy the terminal attitude of the pursuer with respect to the target.

Our sensor network consists of indiscriminately spread data collecting sensors, sensor collecting robot (robot 1), and sensor carrying robot (robot 2) used for sensor localization [8]. This situation is similar to situations where two space vehicles have the requirement to engage with definite terminal characteristics (distance and approach angle with respect to each other) or a missile hitting a target with a desired terminal angle where the principle axis of the missile along which the debris fly, is aligned with a desired direction with respect to the target (on the target). Another instance is in establishing a physical supply line between two vehicles (i.e., on-flight refuelling, loading ammunition etc.).

The need for the terminal attitude guidance problem has been brought about as a result of recent advances in space vehicle engagements, weapon systems and subsystems technologies as well as a shift in guided weapons systems deployment and operational philosophies.

In the past, due to real-time computing constraints in implementing sophisticated algorithms on autonomous systems, major simplifications of engagement kinematics model, performance index constraints had to be implemented in order to render the solution suitable for mechanization of a real system. In this work, we consider achieving a desired terminal attitude of the pursuer with respect to the target. In other words, we need the pursuer’s relative velocity with respect to the target achieves a desired angle and a magnitude at the terminal point of the trajectory while minimizing the miss distance. This is important in general for autonomous space vehicle engagements.

Also here we introduce closed form solutions for the overall controller that optimize the terminal objectives. For simplicity we assume the both objects in the pursuit are point-wise.

Firstly, as in [11], we formulate the terminal attitude guidance problem as a linear-quadratic optimal control problem. The respective controllers can be obtained by the resulting Riccati differential equations. As the main robot (robot 2) is moving with a constant velocity, we can use the optimal control approach. Moreover, two further significant shortcomings in the optimal control approach are, all the states of the pursuit system are precisely known and the initial conditions are known precisely. Further, our problem is even simplified as both robots have GPS and also acceleration command input as well as speedometers enabling us to have access to the state of the dynamic system.

In order to formulate terminal approach guidance laws for our robots, as in [11] engagement kinematics are defined in terms of relative states between the two robots. For the sake of simplicity consider the two dimensional motion of the two robots. The absolute position of robot 1 and robot 2 are \([x_1, x_2]\) and \([x_3, x_4]\) respectively with velocities \([v_{1x}, v_{1y}]\) and \([v_{3x}, v_{3y}]\) respectively. The control input i.e. the acceleration vectors are given as \([a_{1x}, a_{1y}]\) and \([a_{3x}, a_{3y}]\) respectively. Then the autonomous robotic system in the state space form can be written as: Firstly, as we need to control the robot terminal velocity.

\[ x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} v_{1x} - v_{3x} \\ v_{1y} - v_{3y} \\ a_{1x} - a_{3x} \\ a_{1y} - a_{3y} \end{bmatrix} \]  (III.1)
Then the state space model becomes

$$\dot{x} = Ax + B_1u + B_2w$$  \hspace{1cm} \text{(III.2)}$$

where

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$B_2 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Here, $u$ and $w$ are the control input of robot 1 and the uncertain maneuverability (acceleration) of robot 2 respectively.

IV. OPTIMAL CONTROL APPROACH

In this section, ignoring the uncertain maneuverability (for the purpose of optimal control) of the second robot, we suppose that the plant is described by the following linear differential equation

$$\dot{x}(t) = Ax(t) + B_1u(t)$$  \hspace{1cm} \text{(IV.1)}$$

where $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ is the control input. We assume that the initial condition of the system is given,

$$x(0) = x_0$$  \hspace{1cm} \text{(IV.2)}$$

where $x_0 \in \mathbb{R}^n$ is a given vector.

With this system let us associate the performance index

$$J[x(\cdot), u(\cdot)] := \frac{1}{2} \int_0^T (x(T) - h)^T X_T (x(T) - h) + \frac{\alpha}{2} \int_0^T ||u(t)||^2 \, dt.$$  \hspace{1cm} \text{(IV.3)}$$

Here $X_T \geq 0$ is a given matrix, $h \in \mathbb{R}^n$ is a given vector, and $\alpha > 0$ is a given constant.

The linear quadratic optimal control problem can be formulated as follows:

To find the minimum of the functional (IV.3) over the set of all $[x(\cdot), u(\cdot)] \in L_2[0,T]$ satisfying the equations (IV.1) and (IV.2).

$$J[x(\cdot), u(\cdot)] \rightarrow \min.$$  \hspace{1cm} \text{(IV.4)}$$

Introduce the following Riccati differential equation

$$-\dot{S}(t) = A' S(t) + S(t) A - \frac{1}{\alpha} S(t) B_1 B_1' S(t),$$

$$S(T) = X_T.$$  \hspace{1cm} \text{(IV.5)}$$

Furthermore, introduce the following equations

$$-\dot{r}(t) = (A - \frac{1}{\alpha} B_1 B_1' S(t))' r(t), \quad r(T) = X_T h.$$  \hspace{1cm} \text{(IV.6)}$$

$$u_{opt}(t) = -\frac{1}{\alpha} B_1' S(t) x_{opt}(t) + \frac{1}{\alpha} B_1' r(t),$$  \hspace{1cm} \text{(IV.7)}$$

$$-\dot{g}(t) = -\frac{1}{2\alpha} r(t)' B_1 B_1' r(t), \quad g(T) = \frac{1}{2} X_T h.$$  \hspace{1cm} \text{(IV.8)}$$

Now we are in a position to state the following theorem.

**Theorem 4.1:** Consider the linear quadratic optimal control problem (IV.1), (IV.2), (IV.3), (IV.4). Then, for any $x_0$, $h$, $X_T \geq 0$ and $\alpha > 0$, the following statements hold:

(i) The minimum in the linear quadratic optimal control problem (IV.4) is achieved.

(ii) The Riccati differential equation (IV.5) has a unique solution on the time interval $[0,T]$.

(iii) The optimal control law $[x_{opt}(\cdot), u_{opt}(\cdot)]$ is given by the equations (IV.5), (IV.6), (IV.7).

(iv) The optimal cost in the problem (IV.4) is

$$\frac{1}{2} x_0' S(0) x_0 - x_0' r(0) + g(0)$$

where $g(\cdot)$ is defined by (IV.8).

**Proof:** See [6].

V. CLOSING VELOCITY FOR APPROACH ANGLE IMPROVEMENT

If the closing velocity is arbitrary, and only the miss distance and the approach angle is concerned, then we give and explicit expression approach for the technique used in [11]. By taking

$$\hat{h} = \begin{bmatrix} \cos \beta \\ \sin \beta \\ 0 \\ 0 \end{bmatrix}$$  \hspace{1cm} \text{(V.1)}$$

and then $h = \hat{h} c$.

A. Optimal $c$

Here we find the optimal value for the $c$ as far as the performance index is concerned in both LQR control framework. For an arbitrary target/mother craft maneuver, we use the $c = c^o$. If the target is optimally maneuvering, we can use $c = c^H$ or $c^\parallel$ for state or output feedback case respectively.

The optimal cost for the optimal controller from theorem 4.1 is:

$$J_{opt} = \frac{1}{2} x_0' S(0) x_0 - x_0' r(0) + \frac{1}{2} h' X_T h - \frac{1}{2\alpha} \int_0^T r'(t) B_1 B_1' r(t) \, dt.$$  \hspace{1cm} \text{(V.2)}$$

By taking

$$c(t) = \exp \left( -\int_0^T \frac{1}{\alpha} S(t) B_1 B_1' \, dt \right)$$  \hspace{1cm} \text{(V.3)}$$

we notice

$$\frac{\partial}{\partial t} \int_0^T r'(t) B_1 B_1' r(t) \, dt = \left( \int_0^T f(t) B_1 B_1' f(t) \, dt \right) h$$

$$\frac{\partial}{\partial t} r(0) = f'(0).$$  \hspace{1cm} \text{(V.4)}$$

Taking

$$\frac{\partial}{\partial c} J_{opt} =$$

$$\hat{h} \left[ -f'(0) x_0 + \left( X_T - \frac{1}{\alpha} \int_0^T f(t) B_1 B_1' f(t) \, dt \right) \hat{h} \right]$$

\[\text{555}\]
The minimum is achieved when
\[
    c = c^0 = \frac{\beta^* f(0)z_0}{\beta^* \left( X_T - \frac{1}{\alpha} \int_0^T f(t)B_xB_y f(t) \, dt \right) \bar{h}} \tag{V.5}
\]
with
\[
    \frac{\beta^*}{2} \left( 2X_T - \frac{1}{\alpha} \int_0^T f(t)B_xB_y f(t) \, dt \right) \bar{h} > 0 \tag{V.6}
\]

VI. SIMULATIONS AND CONCLUSIONS

Two body pursuit for the sensor network implementation was simulated for the case parameters indicated in table VI for two robots. The robot 1 picks sensors from an arbitrary yet known location and delivers to the robot 2 which is travelling at a constant velocity. At the delivery time, we expect the robots to approach perpendicularly as the access to the sensors accumulation table need to be perpendicular. The distance between the robots in the terminal time is given in figure VI.1 as a function of the parameter \( c \) for the LQR state feedback case. Ideally we expect this to be zero. Figure VI.2 shows the angle between the two robots (approach angle) in the terminal time. For delivery of sensors we expect this to be 90° relative to the accepting robot. As it is evident from these figures, we need to have a compromise on the terminal distance and the approach angle when choosing the value of \( c \). The optimal value derive theoretically is \( c = -0.003 \) (from equation V.5). For this value the miss distance in 8cm and the approach angle is 88.9°. Considering the size of the sensor accumulation table and this accuracy is sufficient for the physical implementation. In this optimal control approach we ignored the maneuverability (acceleration of robot 1). Hence, the derivations are for a constant velocity motions of robot 1. It is easy to extend this to compensate for this using a robust control approach when robot 1 is maneuverable.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T )</td>
<td>Simulation time</td>
<td>16s</td>
</tr>
<tr>
<td>( T_s )</td>
<td>Time step</td>
<td>0.6s</td>
</tr>
<tr>
<td>( \beta )</td>
<td>Desired approach angle</td>
<td>90°</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>Weighting on the control input</td>
<td>0.01</td>
</tr>
<tr>
<td>( x_f(0) )</td>
<td>Initial position of robot 1</td>
<td>[0 200 2 0]</td>
</tr>
<tr>
<td>( y_f(0) )</td>
<td>Initial position of robot 2</td>
<td>[0 0 0 0]</td>
</tr>
</tbody>
</table>

TABLE VI.1

SIMULATION PARAMETERS

REFERENCES


Fig. VI.3. Variation of relative distance of robots for optimal "C"(−0.003)


