БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ • BULGARIAN ACADEMY OF SCIENCES

ПРОБЛЕМИ НА ТЕХНИЧЕСКАТА КИБЕРНЕТИКА И РОБОТИКАТА, **62**
PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS, **62**

София • 2010 • Sofia

Ultrasonic Positioning System for Mobile Robots

Nayden Chivarov, Stefan Shivarov***

** Institute of Systems Engineering and Robotics, 1113 Sofia*
*** Code Assitance Ltd., 1606 Sofia*
E-mails: nshivarov@code.bg shivarov@code.bg

1. Introduction

Positioning systems are widely used today in many fields of applications. A similar system is the Global Positioning System (GPS) [1]. It is a world-wide positioning and navigation system, used for civil, as well as military purposes. It allows comparatively precise determining of the location and route of persons, cars and other vehicles, etc. The precision of the system is good enough for navigation throughout the whole road map of the world and is even better for military purposes.

However, this precision is far from adequate for other purposes, where smaller areas with increased accuracy requirements apply, i.e., robot navigation for industrial or service applications. Such scenarios require their own positioning and navigation system in order to achieve adequate precision in a centimeter or even milimeter range. Several types of systems covering these tasks have been developed today, employing similar techniques to the common GPS system, but implemented either with infrared or ultrasonic waves (or combination of both). Due to the specific requirements in such applications and the search for economically adequate solutions, radio waves are mostly used as a tool for synchronization rather than a primary source of distance/coordinate information.

2. Principles of operation

There are two primary and one auxiliary methods of determining objects' coordinates in relation to a predefined (previously known) set of locations.

2.1. Triangulation

Triangulation is the process of determining the location of a point by measuring the angles to it from known points at either end of a fixed baseline, rather than measuring distances to the point directly (see Fig. 1). The point can then be fixed as the third point of a triangle with one known side and two known angles [2].

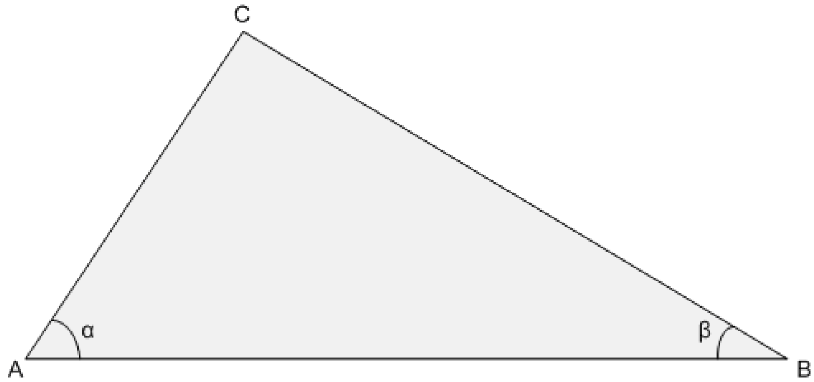

Fig. 1. Triangulation

2.2. Trilateration

Trilateration is a method for determining the intersections of three sphere surfaces given the centers and radii of the three spheres [3].

For the purpose of robot navigation in a small room and two-dimensional plane, the task can be reduced to finding the intersecting points of just two circumferences (Fig. 2).

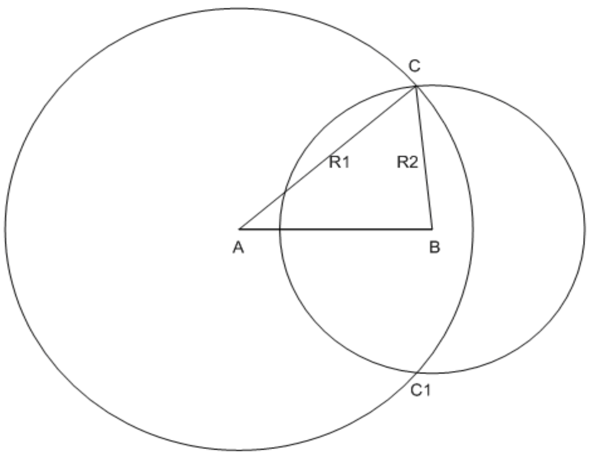

Fig. 2. Reduced trilateration

The solution is needed for a given plain where the mobile objects/robots move. The intersecting points are two – C and C_1, but if the reference points A and B are chosen to be on the same side of the area of navigation then the solution is not ambiguous but is clearly defined. Thus when the distance AB is previously defined, and the two distances R_1 and R_2 are measured, the solution for the coordinates of C is defined and trivial.

This is an easier and cheaper way to find objects' coordinates because the measurement of distances are easily achieved by measuring the time for an ultrasonic wave to travel from two beacons (A and B) to the object (C) or vice versa. Given the speed of sound in air (approximately 340 m/s at 20 ^{o}C), the distances r_1 and r_2 are easily calculated.

2.3. Dead reckoning

This is a supplementary method for achieving better performance when the object is temporary out of range (i.e., behind an obstacle). A very popular example is making use of the encoders attached to robot's wheels and giving the approximate number of counts (that is, the distance) travelled by each of the wheels. This method works very well on short distances and adds overall resilience to the system.

3. Implementation

3.1. Block diagram

The basic block diagram of our Ultrasonic Positioning System for Mobile Robots is shown on Fig. 3. Modules designated as R_xT_x are ultrasonic transceivers. They are implemented with separate coupled US transmitters and receivers (operating in 40 kHz band) in order to reduce the mode switching time (receive/transmit). The transceiver modules have also operational amplifiers and a microcontroller for intelligent control and automated response.

The PM (Processing Module) is synchronized with Object's Transceiver using RF (Radio Frequency) or IR (InfraRed) connection in order to determine the start of R_1, R_2 measurement. As the speed of infrared/radio pulse is much faster than the speed of sound, its propagation time is neglected. The distances R_1 and R_2 are measured by counting the time (in microseconds) needed for the ultrasonic pulse emitted from Object's transceiver to reach PM's Transceivers or vice versa. More about the direction choice is explained below.

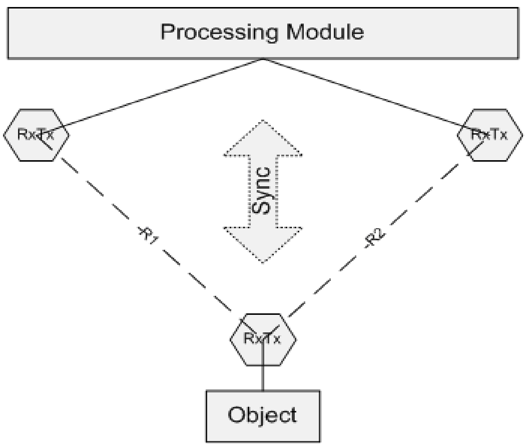

Fig. 3

3.2. Operation

There are two approaches to operating this system, and they are basically equivalent.

- Given a Sync command from the PM, an ultrasonic pulse is emitted from Object's R_xT_x and reaches PMs' R_xT_x-s at times T_1 and T_2. Then the radiuses R_1 and R_2 are calculated:

$$R_i = V_s T_i \,.$$

In this case, the coordinates are calculated in PM and if the object is a robot or a person who needs them, they have to be transmitted through the RF/IR connection from PM to the object.

- Given a Sync command from the PM and sequential ultrasonic pulses from the two PMs' transceivers, the Object's R_xT_x receives the two pulses and counts the times T_1 and T_2. Then distances are calculated using the same formula from above. In this case, the coordinates or at least the raw data (R_1, R_2) are located in the Object's side and if the PM needs them, they have to be transmitted through *Sync* connection.

The differences between the above two approaches are the following:

- The side that acquires the raw data (PM or Object)
- Scalability of the system

The first approach is more appropriate for a system that tracks few objects (i.e., 1-5) while the second approach is better suited for a system that tracks multiple objects simultaneously in real time.

- Objects' speeds

If the objects being tracked are too fast, the second approach is not very accurate as the object moves between the first and the second ultrasonic pulse emitted by PM and thus the second distance (i.e. R_2) is not completely relevant to R_1. In this case a single pulse emitted by the object (and received separately by PMs' transceivers) achieves better relevance and higher precision of the results.

3.3. Algorithms

In order to measure the times of arrival of the ultrasonic signal, the pulses have to be detected. There are two main approaches for implementing US pulse detection:

- Detecting the envelope

Assuming $t = 0$ is the start moment of pulse transmission and U_{th} is a predefined comparator threshold value, the *time-of-travel* T of the ultrasonic pulse is measured as shown on Fig. 4. A simple amplitude detector is used for this purpose. In this case, if certain properties of the environment, such as *reverberation* causes the signal to rise slowly, time T is different from the open-environment case (reverberation free).

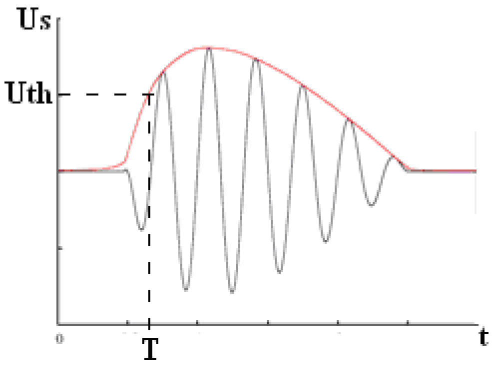

Fig. 4. Measured ultrasonic pulse

- Detecting the signal itself

This is the method used in our ultrasonic position system for mobile robots, because of its inherent resilience to reverberation. The idea is to count *N* pulses from the beginning of the packet and then sample *T*. This also gives some time-domain filtration and makes the system more noise-resilient. The principle is shown on Fig. 5. The number of periods used is *N*=5.

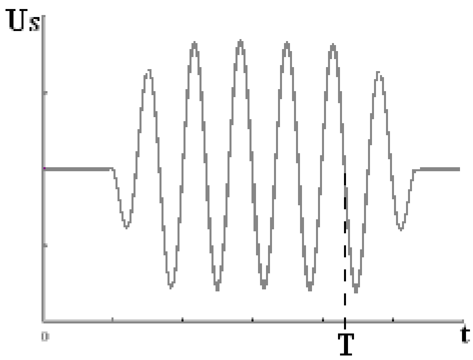

Fig. 5. Detecting the signal itself

3.4. Method timer1_Tick

The main simulation function is the method timer1_Tick, which is called on each Timer Tick message. First, the data of ultrasonic pulses arrival times is received from the device through the serial Port1 connection. The lines that follow calculate and display the playground *x* and *y* coordinates from the two times of arrival. This is done using simple trigonometry, keeping in mind that the two beacons are situated at a certain (known) height over the playground surface. The green dot is displayed on main form's Paint event handler.

```
private void timer1_Tick(object sender, EventArgs e)
{
   byte[] buff = new byte[maxbufflen];
   buff[0] = 1;
```

```
            buff[1] = (byte)v1command;
            buff[2] = (byte)v2command;

            // Communication
             serialPort1.BaseStream.Flush();
            serialPort1.Write(buff, 0, 3);
            serialPort1.Read(buff, 0, 4);

            // Calculations
            Lengths.x = buff[0] + (buff[1] << 8) - heightCorrection;
            Lengths.y = buff[2] + (buff[3] << 8) - heightCorrection;
            toolStripStatusLabel1.Text = Lengths.x.ToString() + ", " +
Lengths.y.ToString();

            double temp;
            temp = (l21 * l21 - l11 * l11 + Lengths.x * Lengths.x - Lengths.y *
Lengths.y) / (2 * Math.Sqrt(l21 * l21 - l11 * l11));
            if(temp > 0)
               robotLocation.x = temp;
            else
               robotLocation.x = 0;
            temp = Math.Sqrt(Lengths.x * Lengths.x - l11 * l11 - robotLocation.x *
robotLocation.x);
            if(temp > 0)
               robotLocation.y = (float)temp;
            else
               robotLocation.y = 0;
            robotLocationGrid.x = robotLocation.x / fieldGridSize.x * Width;
            robotLocationGrid.y = Height - robotLocation.y / fieldGridSize.y * Height;
            toolStripStatusLabel2.Text = ((UInt32)robotLocationGrid.x).ToString() + ",
" + ((UInt32)robotLocationGrid.y).ToString();
            Refresh();
            if (first != 0)
            {
               robotLocationPrev.x = robotLocation.x;
               robotLocationPrev.y = robotLocation.y;
               robotLocationGridPrev.x = robotLocationGrid.x;
               robotLocationGridPrev.y = robotLocationGrid.y;
               first--;
            }
//             if (++ct >= 5)
            {
               ct = 0;
               robotVelocity.x = robotLocation.x - robotLocationPrev.x;
               robotVelocityGrid.x = robotLocationGrid.x - robotLocationGridPrev.x;
               robotVelocity.y = robotLocation.y - robotLocationPrev.y;
               robotVelocityGrid.y = robotLocationGrid.y - robotLocationGridPrev.y;
```

```
            robotHeading = robotVelocityGrid;
            toolStripStatusLabel3.Text = "vx = " + ((int)(robotVelocity.x)).ToString()
+ ", vy = " + ((int)(robotVelocity.y)).ToString();
            robotLocationPrev.x = robotLocation.x;
            robotLocationGridPrev.x = robotLocationGrid.x;
            robotLocationPrev.y = robotLocation.y;
            robotLocationGridPrev.y = robotLocationGrid.y;
        }

        robotLocationGridNext = robotLocationGrid + robotVelocityGrid;
        if (xtarg != 0 || ytarg != 0)
        {
            destination = target - robotLocation;
            alpha = Math.Atan2(destination.y, destination.x) -
Math.Atan2(robotVelocity.y, robotVelocity.x);
            if (alpha < 0)
            {
                v1command = 50;
                v2command = 30;
            }
            if (alpha > 0)
            {
                v1command = 30;
                v2command = 50;
            }
            if(alpha == 0)
            {
                v1command = 30;
                v2command = 30;
            }
            if(Math.Abs(xnext - xtarg) < 100 && Math.Abs(ynext - ytarg) < 100)
            {
                v1command = 0;
                v2command = 0;
            }

        }
        toolStripStatusLabel4.Text = ((int)(robotLocation.dispx())).ToString() + ", "
+ ((int)(robotLocation.dispy())).ToString();
    }
```

4. Limitations

There are several sources of limitation of implementations like the one described in this article.

The first one is the properties of the signal itself. The ultrasonic pulse has a central frequency of 40 kHz that is standard for most ultrasonic applications used today. Due to the diffraction properties of small objects, using such waves limits the precision to the order of wavelength λ which for the given frequency is calculated:

$$\lambda = \frac{V_s}{f_s} \approx \frac{340\ \text{m}}{40000} = 8.5\ \text{mm}.$$

The second limitation comes from the properties of the environment. Usually there are lots of reflections from surrounding objects, floor, ceiling, etc. They create interference with the primary signal and sometimes it may prove difficult to distinguish between them. The *reverberation* is yet another property (for closed environments) that may cause the signal level to rise and fall slowly thus leading to the need of calibration of the system for each particular environment, even if *AB* distance and the height of the beacons is the same.

5. Applications

The basic principles described in this article, as well as the method and algorithms can be used in a vast variety of scenarios and solve many different tasks. Starting with amusement applications (games, automated robot play) and continuing to the fields of advertising, automated guides for museums, etc., and continuing to many industrial applications, automated performance of different tasks in a common environment, home appliances and robotics, artificial intelligence systems, navigation in buildings, complex environments, etc. Each modern task, requiring navigation can benefit from such a system (Fig. 6).

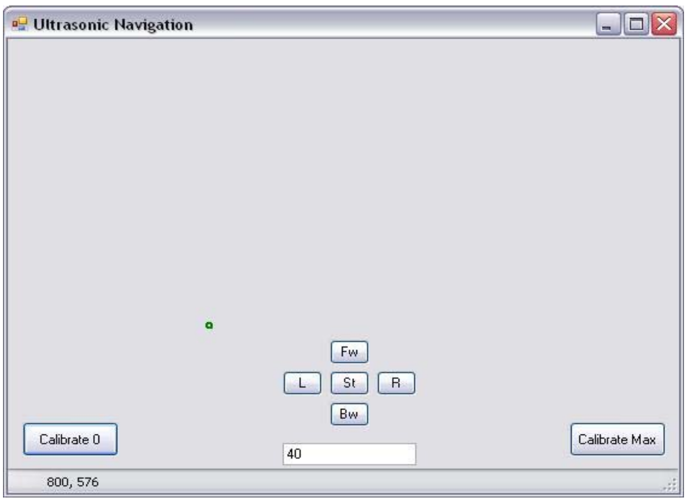

Fig. 6. Our ultrasonic navigation system for a mobile robot

6. Future development

The future development of this project can be realized in several directions:

- Improving performance – reducing errors, increasing precision and stability, range, etc.;
- Increasing the fields of application – creating and improving innovative solutions, adding intelligent control to the existing ones;
- Embedding of the navigation system into existing robot applications – merging data from various sensors and decision making is a challenging task and contributes to the field of Artificial Intelligence.

References

1. **http://en.wikipedia.org/wiki/Global_Positioning_System**
2. **http://en.wikipedia.org/wiki/Triangulation**
3. **http://en.wikipedia.org/wiki/Trilateration**
4. **http://www.cs.uwyo.edu/~wspears/papers/iros07.pdf**
5. **http://nms.lcs.mit.edu/projects/cricket/**
6. **http://www.sonitor.com/**
7. **http://www.gc.vgtu.lt/upload/geod_zurn/geodezija 2009 1 03 mautz.pdf**
8. T h o m a s, F., L. R o s. Revisiting Trilateration for Robot Localization. – IEEE Tranzaction on Robotics, **21**, 2005, No 1, 93-101.
9. N i s s a n k a, B. P. The Cricket Indoor Location System. Doctoral Thesis. Massachusetts Institute of Technology, Cambridge, MA 2005.
10. P e a s g o o d, M., C. C l a r k, J. M c P h e e. Localization of Multiple Robots with Simple Sensors. – In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS’05, 2005, 671-676.

Ультразвуковая позиционная система для мобильных роботов

Найден Шиваров, Стефан Шиваров***

** Институт системного инженерства и робототехники, 1113 София*
*** „Код Асистенс” АД, 1606 София*
E-mails: nshivarov@code.bg shivarov@code.bg

(Р е з ю м е)

В настоящее время разрабатываются несколько типов позиционных и навигационных систем для промышленных и сервисных роботов, используя технические способы, подобные Глобальной позиционной системе (GPS), но они осуществляються либо инфракрасными, либо ультразвуковыми волнами. Наша ультразвуковая позиционная система для мобильных роботов достигает адекватной точности даже в сантиметровом или в миллиметровом диапазоне.

Процессный модуль этой специализированной позиционной системы с ультразвуковыми трансиверами для мобильных роботов синхронизован с трансивером (приемо-передатчиком) объектов, используя радиочастоты.

Разработанная нами ультразвуковая позиционная система можно применять в большом разнообразии сценариев. С её помощью можно решать множество разнообразных задач, начиная с навигацией сервисных и персоналных роботов (в том числе роботов игрушек, и т. д.), а также она находит применение и в многих отраслях промышлености.