Copyright © 2002 IFAC
15th Triennial World Congress, Barcelona, Spain

DISTRIBUTED POSITION ESTIMATION FOR SENSOR NETWORKS [⋆]

Jun Zhang and Shankar Sastry

Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720
{zhangjun, sastry}@eecs.berkeley.edu

Abstract: In this paper we study distributed position estimation for sensor networks. The fundamentals of distributed position estimation algorithms are presented. Two types of ellipsoid outer-approximation algorithms are employed to estimate the positions of unknown sensors. Polytope outer-approximation algorithm is also investigated. Numerical experiments demonstrate the effectiveness of the algorithms. It turns out that polytope type of algorithm is able to yield position estimations more accurately and efficiently.

Keywords: Sensor networks, Distributed estimation, Outer-approximation

1. INTRODUCTION

Sensor network has become an area of great research interest recently. Dramatic advances in MEMS, digital circuitry, and wireless communication technology have enabled us to build sensors with smaller size, lower cost and less power consumption. A massive number of these sensors can be easily deployed into an environment to make up self-calibrating and disposable sensor networks. We envision numerous applications for such networks. Examples include weather monitoring, military surveillance, and environmental exploration. In sensor networks, both local and distributed inference algorithms can be employed to interpret the measured data at multiple levels of granularity, and those interpretations can be circulated in response to events or queries. With these features, sensor networks are ideal in monitoring and exploring a wide range of environment at a reasonable cost. Current research in this area includes SmartDust at Berkeley (Kahn *et al.*, 2000), CoSense at Xerox PARC (Chu *et al.*, 2002), and SenserWebs at JPL (JPL, 2001).

In this paper, we develop distributed outer-approximation algorithms for position estimation in sensor networks. When thousands of sensors are scattered throughout an environment, the geographical distribution of the sensors will be initially unknown and will depend on both the scattering process and the physical structure of the environment. One key task is to determine the spatial localization of the sensors. Sensors could be equipped with Global Positioning System (GPS) units to acquire their accurate position information; however, this is not appealing due to the high cost and large power consumption. One trade-off solution is to estimate the positions of unknown sensors in a cooperative and distributed manner. In such a network, only a few GPS-equipped sensors (beacons) know their own positions, and all the other unknown sensors could estimate their own positions from the knowledge about the beacons in their neighborhood. A similar problem is the localization of distributed robotic team, see (Kleeman, 1992), (Leonard and

[⋆] This work was supported in part by DARPA F30602-00-2-0538 and ARO under the MURI grant DAAH04-96-1-0341.

Durrant-Whyte, 1991), and (Navarro-Serment *et al.*, 1999).

The outline of the paper is as follows. Section 2 presents the basic idea of distributed position estimation. In section 3, two types of ellipsoid outer-approximation algorithms are given to estimate the positions of unknown sensors. In section 4, polytope outer-approximation algorithm is used to accomplish the position estimation. Section 5 presents numerical experiments to demonstrate the effectiveness of the algorithms. A conclusion is given in Section 6.

2. DISTRIBUTED POSITION ESTIMATION

Suppose that each sensor in a sensor network has an on-board communication module so that it can establish local communication connectivity with a set of neighboring sensors. If an unknown sensor is able to receive communication signals from a nearby beacon, it must lie in a disc centered at that beacon with the radius of the maximum communication range R. Moreover, if this sensor can receive the position information of m beacons in its neighborhood, it has to lie in the intersection of all these m discs. Therefore, an outer-approximation of this intersection could be used as an estimation of the position of the unknown sensor. Every unknown sensor is capable of performing position estimation algorithms with its own computational power by using the received accurate positions of its neighboring beacons, and the estimated position can be stored in its own memory. The position estimations of the whole sensor network can thus be done in such a distributed fashion.

Now the main problem is to find algorithms to outer-approximate the intersection of m discs. The algorithms have to be numerically efficient and tight in order to be implemented on sensors with only limited computational ability and yet to provide an accurate estimation. In (Doherty, 2000), the rectangular outer-approximation is explored via semidefinite programming. We develop the ellipsoid and polytope outer-approximation algorithms in this paper. Refer to (Kurzhanski and Vályi, 1996) for more details on ellipsoidal calculus, (Tikhomirov, 1990) and (Tuy, 1998) on convex analysis.

3. ELLIPSOID OUTER-APPROXIMATION

In this section, ellipsoid is used to outer-approximate the intersection of m discs. Position estimation algorithms are performed in a sequential manner. To be more specific, we first find a series of circumscribed ellipsoids to cover the intersection of disc i and $i+1$, where $i = 1, \ldots, m-1$. Then, for these $m-1$ ellipsoids, we utilize a new series of ellipsoids to outer-approximate the intersection of the ellipsoids i and $i+1$, where $i = 1, \ldots, m-2$. By iterating this procedure $m-1$ times, we can finally obtain a single ellipsoid that outer-approximates the intersection of all the m discs. Due to the nature of the iteration, the unknown sensor must lie in the final ellipsoid. The advantage of the sequential outer-approximation procedure is that it avoids to deal with all the m discs simultaneously, which significantly reduces the computational loads.

An ellipsoid with center a and configuration matrix Q is denoted as

$$\mathcal{E}(a,Q) = \{x \in \mathbb{R}^2 \mid (x-a)^T Q^{-1}(x-a) \leq 1\}.$$

In this paper, all the configuration matrices are assumed to be non-degenerate.

3.1 Outer-approximation of the intersection of two discs

For the position estimation algorithms based on ellipsoid outer-approximation, we need to consider two types of outer-approximation problems: one is the computation for the intersection of two discs, the other is for two ellipsoids. Theoretically these two types of problems are the same due to the obvious fact that a disc is also an ellipsoid. However, we would rather treat them separately because the ellipsoids obtained to outer-approximate the intersection of two ellipsoids are usually not tight (Kurzhanski and Vályi, 1996), whereas a tight ellipsoid can be calculated in a closed form to outer-approximate the intersection of two discs.

Let us consider two discs $B(b,R)$ and $B(c,R)$. There exists a unique circumscribed ellipsoid $\mathcal{E}(a,Q)$ to outer-approximate their intersection (See Figure 1), where

$$Q = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \times \begin{bmatrix} \left(R - \frac{\|c-b\|}{2}\right)^{-2} & 0 \\ 0 & \left(R^2 - \frac{\|c-b\|^2}{4}\right)^{-1} \end{bmatrix} \times \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix},$$

and

$$(\cos\theta, \sin\theta)^T = \frac{c-b}{\|c-b\|}, \qquad a = \frac{b+c}{2}.$$

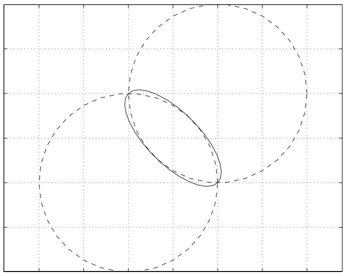

Fig. 1. Ellipsoid outer-approximation of the intersection of two discs.

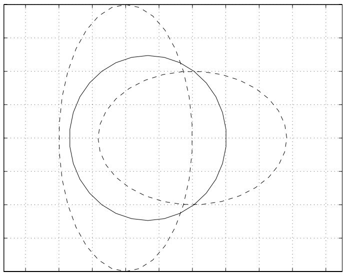

Fig. 2. Outer-approximation of the intersection of two ellipsoids: Algorithm I.

Thus we obtain a collection of $m-1$ ellipsoids $\mathcal{E}(a_i, Q_i)$.

3.2 *Outer-approximation of the intersection of two ellipsoids: Algorithm I*

Now we consider the outer-approximation of the intersection of two ellipsoids $\mathcal{E}(a_1, Q_1)$ and $\mathcal{E}(a_2, Q_2)$. We present two different algorithms in this paper.

Definition 1. Given two convex compact sets $\mathcal{H}_1$, $\mathcal{H}_2 \in \mathbb{R}^n$, the *geometrical (Minkowski) sum* of $\mathcal{H}_1$ and $\mathcal{H}_2$ is defined as

$$\mathcal{H}_1 + \mathcal{H}_2 = \cup_{h_1 \in \mathcal{H}_1} \cup_{h_2 \in \mathcal{H}_2} \{h_1 + h_2\}.$$

Proposition 1. (Kurzhanski and Vályi, 1996) The intersection P of m non-degenerate ellipsoids $\mathcal{E}(a_i, Q_i)$ satisfies the following equality

$$P = \bigcap \sum_{i=1}^{m} \mathcal{E}(M_i a_i, M_i Q_i M_i^T),$$

where $\sum_{i=1}^{m} M_i = I$.

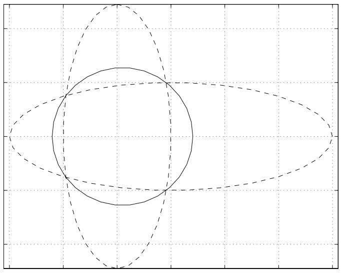

Fig. 3. Outer-approximation of the intersection of two ellipsoids: Algorithm II.

In particular, when $m = 2$ we have

$$\begin{aligned} &P \subseteq \mathcal{E}_1 + \mathcal{E}_2, \\ &\mathcal{E}_1 = \mathcal{E}(Ma_1, MQ_1M^T), \\ &\mathcal{E}_2 = \mathcal{E}((I-M)a_2, (I-M)Q_2(I-M)^T), \end{aligned}$$

where $M \in \mathbb{R}^{2\times 2}$. Further, the geometrical sum of $\mathcal{E}_1$ and $\mathcal{E}_2$ can be approximated externally by a new ellipsoid, namely,

$$\mathcal{E}_1 + \mathcal{E}_2 \subseteq \mathcal{E}(a[M], Q[p, M]),$$

where

$$\begin{aligned} a[M] &= Ma_1 + (I-M)a_2, \\ Q[p, M] &= (1+p^{-1})MQ_1M^T + \\ &\qquad (1+p)(I-M)Q_2(I-M)^T. \end{aligned}$$

We then pick

$$M^* = (Q_1 + Q_2)^{-1} Q_2$$

to minimize the sum of $\mathrm{tr} MQ_1M^T$ and $\mathrm{tr}(I - M)Q_2(I-M)^T$, and

$$p^* = \sqrt{\mathrm{tr} M^* Q_1 {M^*}^T / \mathrm{tr}(I - M^*)Q_2(I-M^*)^T}$$

to minimize

$$\begin{aligned} \mathrm{tr}\, \Big(&(1+p^{-1})M^*Q_1{M^*}^T + \\ &(1+p)(I-M^*)Q_2(I-M^*)^T \Big). \end{aligned}$$

Therefore, we have

$$\begin{aligned} P \subseteq \mathcal{E}\Bigg(&M^*a_1 + (I-M^*)a_2, (1 + \frac{1}{p^*})M^*Q_1{M^*}^T \\ &+ (1+p^*)(I-M^*)Q_2(I-M^*)^T \Bigg). \end{aligned}$$

Generally this algorithm does not yield tight outer-approximating ellipsoids as shown in Figure 2. Here a tight outer-approximating ellipsoid means that we cannot find a new outer-approximating ellipsoid with smaller area by shrinking either of its semiaxes. For an outer-approximating ellipsoid obtained from algorithm I, we can usually decrease its semiaxes in order to find a tight outer-approximating ellipsoid.

3.3 *Outer-approximation of the intersection of two ellipsoids: Algorithm II*

We present another algorithm to outer-approximate the intersection of two ellipsoids in this subsection. Define

$$\mathcal{A} = \left\{ \alpha \in \mathbb{R}^2 \,\middle|\, \alpha_1 + \alpha_2 = 1, \alpha_1 \geq 0, \alpha_2 \geq 0 \right\},$$

and notice the inequality

$$\alpha_1 (x - a_1)^T Q_1^{-1} (x - a_1) + \\ \alpha_2 (x - a_2)^T Q_2^{-1} (x - a_2) \leq 1.$$

It is not difficult to observe that for a given $\alpha \in \mathcal{A}$ the above inequality defines an ellipsoid

$$\mathcal{E}(\alpha) = \{x \in \mathbb{R}^2 \mid (x - a[\alpha])^T \\ Q^{-1}[\alpha](x - a[\alpha]) \leq 1 - h[\alpha]\},$$

where

$$\begin{aligned} Q[\alpha] &= \left(\alpha_1 Q_1^{-1} + \alpha_2 Q_2^{-1}\right)^{-1}, \\ h[\alpha] &= \alpha_1 a_1^T Q_1^{-1} a_1 + \alpha_2 a_2^T Q_2^{-1} a_2 - \\ &\quad (\alpha_1 Q_1^{-1} a_1 + \alpha_2 Q_2^{-1} a_2)^T Q[\alpha] \\ &\quad (\alpha_1 Q_1^{-1} a_1 + \alpha_2 Q_2^{-1} a_2), \\ a[\alpha] &= Q[\alpha] \left(\alpha_1 Q_1^{-1} a_1 + \alpha_2 Q_2^{-1} a_2\right). \end{aligned}$$

By using Cauchy-Schwartz inequality, we can prove that $h[\alpha] \in [0, 1]$. Therefore, we have that

$$\mathcal{E}(\alpha) = \mathcal{E}(a[\alpha], (1 - h[\alpha])Q[\alpha]).$$

It is straightforward to prove the following assertion:

$$P = \bigcap_{\alpha \in \mathcal{A}} \mathcal{E}(\alpha)$$

The intersection of two ellipsoids is now outer-approximated by a parameterized family of ellipsoids $\{\mathcal{E}(\alpha) \mid \alpha \in \mathcal{A}\}$. We want to find an $\alpha^* \in \mathcal{A}$ such that $\mathcal{E}(\alpha^*)$ contains the minimum area. This can be done by a standard nonlinear optimization solver. Note that $\mathcal{E}(\alpha)$ actually defines a homotopy which continuously deforms $\mathcal{E}(a_1, Q_1)$ into $\mathcal{E}(a_2, Q_2)$: when $\alpha = (1, 0)^T$, $\mathcal{E}(\alpha) = \mathcal{E}(a_1, Q_1)$; and when $\alpha = (0, 1)^T$, $\mathcal{E}(\alpha) = \mathcal{E}(a_2, Q_2)$.

Algorithm I can yield a tight outer-approximating ellipsoid when two ellipsoids intersect at four points. However, when two ellipsoids intersect at only two or three points, usually the resulting ellipsoid is not tight. Algorithm II usually yields a better outer-approximation solution than Algorithm I in this case, but it also demands more computational power since it needs to solve a nonlinear optimization problem. The advantage of Algorithm I is that it has a closed form solution, hence it is much more numerical efficient.

4. POLYTOPE OUTER-APPROXIMATION

In this section we use polytope to outer-approximate the intersection of m discs. The procedure is also accomplished in a sequential manner, which is similar to that in the ellipsoid outer-approximation algorithms.

A convex set is said to be *polyhedron* if it is the intersection of a finite family of closed half-spaces. In other words, a polyhedron is the solution set of a finite number of linear inequalities in the form

$$(a^i, x) \leq b_i, \quad i = 1, \ldots, m, \tag{1}$$

where a^i, $x \in \mathbb{R}^n$, and $b_i \in \mathbb{R}$. It can also be represented in a matrix form

$$Ax \leq b,$$

where A is a $m \times n$ matrix of rows a^i and $b = (b_1, \ldots, b_m)^T$. An inequality $(a^k, x) \leq b_k$ is said to be *redundant* if the removal of this inequality from (1) does not affect the polyhedron, i.e., if the system (1) is equivalent to

$$(a^i, x) \leq b_i, \quad i \in \{1, \ldots, m\} \setminus \{k\}.$$

A bounded polyhedron is called a *polytope*.

The intersection of two discs can be outer-approximated by a polytope as in Figure 4. As in the ellipsoid case, assume that in the neighborhood of an unknown sensor there are m beacons, we can thus obtain $m - 1$ outer-approximating polytopes. Now we compute the intersection of these polytopes. Consider two polytopes

$$D_1 = \{x \in \mathbb{R}^2 \mid A_1 x \leq b^1\}, \\ D_2 = \{x \in \mathbb{R}^2 \mid A_2 x \leq b^2\},$$

their intersection is simply a new polytope

$$P = \left\{x \in \mathbb{R}^2 \mid \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} x \leq \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}\right\}. \tag{2}$$

Since the unknown sensor lies in the intersection of polytopes D_1 and D_2, polytope P is obviously non-empty. System (2) may contain redundancy, hence we need to remove it from the system. For any inequality $(a^k, x) \leq b_k$ in (2), if there exist exactly two adjacent vertices of polytope P on the line $\{x \in \mathbb{R}^2 \mid (a^k, x) = b_k\}$, then it is not redundant; otherwise, it has to be removed from the system. Technical details on how to find the vertices of a polytope are omitted here.

Recall that in the ellipsoid case, we need to find a new ellipsoid to outer-approximate the intersection of two existing ellipsoids. However, we don't need to do that in the polytope case. The intersection of two polytopes is just another new polytope, and it can be efficiently computed. There is no new approximation errors introduced in the iteration procedure. One potential problem for polytope outer-approximation could be that as the iteration evolves, the numbers of edges and vertices of the outer-approximating polytope may

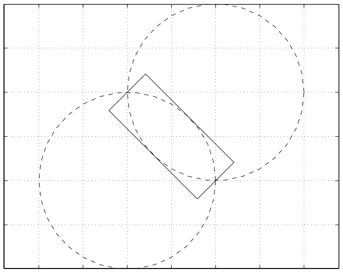

Fig. 4. Polytope outer-approximation of the intersection of two discs.

increase to a large number, which may impose a burden to the limited local memory on the sensors.

Note that by now we have only made use of the position information from beacons. After an unknown sensor acquires an estimation of its own position, this information can be used to refine the initial estimations for all the other unknown sensors in its neighborhood. To be more specific, consider two unknown sensors S_1 and S_2. Suppose that S_1 lies within the maximum communication range of S_2, and both of the sensors have initial estimation polytopes denoted as D_1 and D_2 respectively. We know that S_1 must lie in a disc centered at S_2 with the radius R. Since S_2 can lie anywhere in polytope D_2, we expand D_2 omnidirectionally by R to a new polytope D_2'. Given the estimation polytope D_2 of S_2, D_2' is actually a polytope which contains all the possible positions of the unknown sensor S_1. Therefore, if we intersect D_2' with the initial estimation polytope D_1 of S_1, a more precise estimation polytope can be obtained for S_1. The same procedure can be applied to all the unknown sensors in the network.

5. NUMERICAL EXPERIMENTS

In this section, we present some numerical experiments on the distributed position estimation algorithms aforementioned. We use the centers of the final approximation ellipsoids or polytopes as the position estimations of unknown sensors.

Consider a randomly generated sensor network with 200 sensors, and suppose that 100 randomly picked sensors know their own positions. Our task is to estimate the positions of the remaining 100 unknown sensors. The sensors are deployed in the area $[0, 10] \times [0, 10]$ with the maximum communication range 1. We apply both ellipsoid and polytope outer-approximation algorithms to estimate the positions of unknown sensors. The results from ellipsoid algorithm and polytope algorithm are shown in Figure 5.

Now we compare the performance of ellipsoid and polytope algorithms. The performance index is defined as:

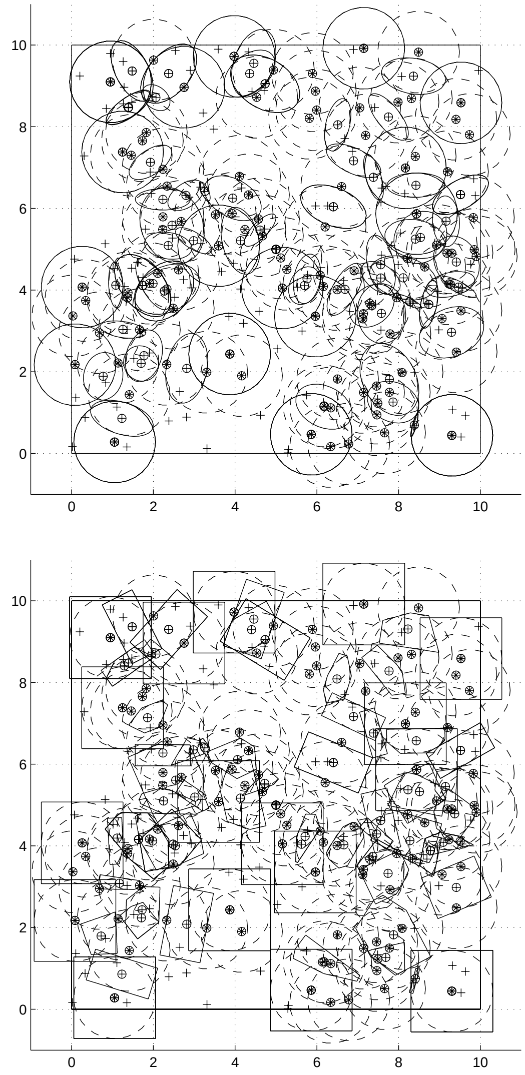

Fig. 5. Ellipsoid and polytope outer-approximation result. Legend: encircled star: beacon; dashed circle: communication range of beacon; plus sign: unknown sensor; encircled plus sign: estimation of unknown sensor; solid ellipsoid: outer-approximation ellipsoid; solid polytope: outer-approximation polytope.

$$\text{err} = \frac{1}{M} \sum_{i=1}^{M} \|x^i_{\text{est}} - x^i_{\text{real}}\|^2, \qquad (3)$$

where M is the number of the sensors whose position can be estimated. To evaluate the algorithms accurately, we disregard the data from those sensors with no nearby beacons. Take 100 randomly generated sensor networks, each of which consists of 200 sensors, including 20 randomly picked beacons. Compute the performance index for each sensor network, and then average it over all the 100 networks. Increasing the number of beacons, we finally obtain the comparison result as shown in Figure 6. It is clear that the mean square error decreases as the number of beacons increases, which is consistent with the intuition. Moreover, the performance of polytope approximation is bet-

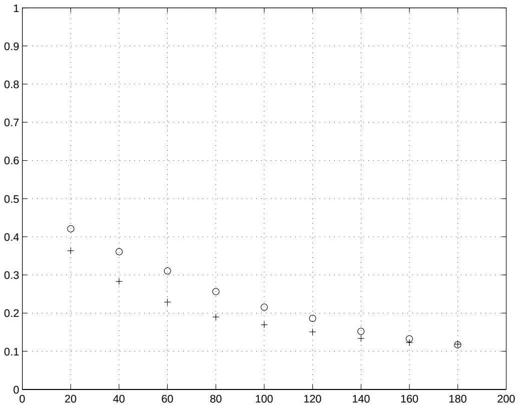

Fig. 6. Performance comparison between ellipsoid and polytope outer-approximations: Plus sign: average mean square error of polytope approximation; circle: average mean square error of ellipsoid approximation. x axis: number of beacons; y axis: mean square error.

ter than that of ellipsoid approximation. Therefore, we would prefer polytope type of position estimation algorithm since it provides more accurate estimations and is more numerically efficient.

6. CONCLUSION

Distributed position estimation algorithms in sensor networks have been presented. Both ellipsoid and polytope approaches are used to outer-approximate the intersection of a finite number of discs, which in turn yield position estimations for unknown sensors in the network. Numerical experiments demonstrate the effectiveness of the algorithms, and polytope type of position estimation algorithm turns out to be a more accurate and efficient approach. We only consider the planar sensor networks in this paper; however, the algorithms here can be readily extended to three dimensional case. We expect to implement the algorithms on the real sensor networks in the near future.

REFERENCES

Chu, M., H. Haussecker and F. Zhao (2002). Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. In: *Int. J. High Performance Computing Applications, to appear.*

Doherty, Lance (2000). Algorithm for position and data recovery in wireless sensor networks. Master's thesis. UC Berkeley.

JPL (2001). *JPL Sensor Webs Project, http://sensorwebs.jpl.nasa.gov/index.html.* NASA Jet Propulsion Laboratory.

Kahn, J. M., R. H. Katz and K. S. J. Pister (2000). Emerging challenges: Mobile networking for smart dust. *Journal of Communications and networks.*

Kleeman, L. (1992). Optimal estimation of position and heading for mobile robots using ultrasonic beacons and dead-reckoning. In: *Proc. of the 1992 IEEE Int. Conf. on Robotics and Automation.* pp. 2582–2587.

Kurzhanski, Alexander and István Vályi (1996). *Ellipsoidal Calculus for Estimation and Control.* Birkhäuser.

Leonard, J. J. and H. F. Durrant-Whyte (1991). Mobile robot localization by tracking geometric beacons. *IEEE Trans. on Robotics and Automation* **7**(3), 376–382.

Navarro-Serment, L. E., C. J. J. Paredis and P. K. Khosla (1999). A beacon system for the localization of distributed robotic teams. In: *Proc. of the Int. Conf. on Field and Service Robotics.* Pittsburgh, PA.

Tikhomirov, V. M. (1990). *Analysis II.* Vol. 14 of *Encyclopaedia of Mathematical Sciences.* Springer-Verlag.

Tuy, Hoang (1998). *Convex Analysis and global optimization.* Kluwer Academic Publishers.