

## Local Positioning with Parallelepiped Moving Grid

Niilo SIROLA and Simo ALI-LÖYTTY

Tampere University of Technology, Finland, e-mail: (niilo.sirola, simo.ali-loytty)@tut.fi

**Abstract** - One of the challenges in personal positioning is the efficient numerical solution of the nonlinear filtering equations resulting from the fusion of different measurement sources. One approach to nonlinear filtering is based on approximating probability density functions with piecewise constant functions. In this work, a moving grid filter is outlined and it is applied to the hybrid local positioning problem. Using a linear motion model and a particular uniformly spaced parallelepiped grid design makes the computation load of the method feasible. Numerical test results from a Matlab implementation are presented, and it is found that the grid method is fast enough for real-time implementation.

### 1 Introduction

One of the challenges in personal positioning is to provide accurate or at least correct position information in urban areas or indoors. The behaviour of satellite-based systems such as GPS is unpredictable at best when used indoors in high-sensitivity mode. Local wireless networks, such as the cellular network, WLAN [1], or Bluetooth [2], offer some positioning capacity but with accuracy inferior to GPS. Another possible component of a personal navigator are the on-board sensors such as accelerometers, barometers or digital compasses.

Combining the various measurement sources is difficult because of different error characteristics, unpredictable distortions, or systematic errors in measurements, strong nonlinearity, complex time-dependencies, and missing data. It is not simple to explicitly model all the cases in a general way, let alone solve the models accurately. Even with correct models, the commonly used Kalman filter and its nonlinear extensions can fail without warning [3].

A recursive Bayesian estimator propagates the complete state distribution and thus can represent also multimodal or under-determined systems accurately. In the Bayesian framework [4, 5], measurement errors do not have to be Gaussian but any measurement model that best matches the real-world situation can be used. The recursive Bayesian filter is most often realised as a sequential Monte Carlo filter, also known as particle filter [6]. The open issue with particle filters is how to find a “sufficient” number of particles in order for the particle swarm to represent the densities correctly.

This work presents an alternative approach to Bayesian filtering based on piecewise constant approximations of the probability densities and an application in hybrid local positioning. Furthermore, we show how using a particular uniformly spaced parallelepiped grid design makes the computation load of the method feasible.

#### 1.1 The nonlinear filtering problem

Consider the discrete-time system with  $d$ -dimensional state  $x_k$  and measurements  $y_k$ :

$$x_{k+1} = f_k(x_k) + w_k, \quad x_k, w_k \in \mathbb{R}^d, \quad f_k : \mathbb{R}^d \rightarrow \mathbb{R}^d, \quad (1)$$

$$y_k = h_k(x_k) + v_k, \quad y_k, v_k \in \mathbb{R}^{m_k}, \quad h_k : \mathbb{R}^d \rightarrow \mathbb{R}^{m_k}. \quad (2)$$

Equation (1) is called the *motion model* and (2) the *measurement model*. The number of measurements  $m_k$  need not be same on every time instant.

For abbreviated notation, mark the probability density function (pdf) of the state  $x_k$  given measurements up to time  $l$  with

$$p_{k|l}(x_k) = p(x_k \mid y_1, y_2, \dots, y_l).$$

Let the probability of measurement  $y$  given position  $x$ , called the *likelihood function*, be

$$L_k(x_k) = p(y_k \mid x_k) = p_{v_k}(y_k - h(x_k)),$$

and the *transition pdf*

$$\phi_k(x_{k+1} \mid x_k) = p_{w_k}(x_{k+1} - f_k(x_k)).$$

The algorithm for recursive Bayesian estimation is:

**Step 0.** Begin with **initial distribution**  $p_{0|0}$ . Set  $k = 1$ .

**Step 1.** Compute the **predictive pdf**

$$p_{k|k-1}(x) = \int_{\mathbb{R}^d} \phi_{k-1}(x \mid \xi) p_{k-1|k-1}(\xi) d\xi. \quad (3)$$

**Step 2.** Compute the **posterior pdf**

$$p_{k|k}(x) \propto p_{k|k-1}(x) L_k(x). \quad (4)$$

**Step 3.** Output posterior mean  $E(x_k)$  and variance  $V(x_k)$ .

**Step 4.** Increment  $k$  and repeat from Step 1.

Analytical solutions to the Bayesian filtering problems are known only for a few special cases, most notably the linear Gaussian model for which the Kalman filter is an exact Bayesian solution. There are also some special saturation-type processes for which the Benes filter is exact [7]. For almost all real-world problems, however, the intermediate distributions have to be approximated numerically.

## 2 Personal positioning problem

In the personal positioning setting, the most common motion model is the position-velocity model, where the state consists of three-dimensional position and velocity:

$$x_k = \begin{bmatrix} r_k \\ v_k \end{bmatrix}$$

and state transition function in (1), when the time interval is one second, is

$$f_k(x) = \begin{bmatrix} I_{3 \times 3} & I_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} x.$$

We will work with position estimates derived from three different types of measurements: range, range difference and planar. The subscript  $k$  identifying the time instant is omitted for clarity except where necessary.

Given the true position  $r$ , a *range measurement* to a station at position  $s$  can be written as

$$h(r) = \|s - r\|.$$

The associated measurement error  $v$  need not be normal. Ideally, we would use an empirically determined distribution that matches the real situation, as is done with short-range wireless networks, for example [2].

The biased range measurements that are typical in GPS are treated as range differences. One of the stations is chosen as reference station and all the differences are formed with respect to it. Let the reference station be at  $s_0$ , then the *range difference measurement* is

$$h(r) = \|s - r\| - \|s_0 - r\|.$$

Note that if there are several range difference measurements, their errors are correlated because all share the error in the reference range. This has to be taken into account when constructing the error covariance matrix.

Finally, the *planar measurement* is

$$h(r) = u^T r,$$

where  $u$  is a unit vector.

On each time step, stack all available measurements into a vector  $y_k$  and the corresponding measurement equations into a vector function  $h_k(x)$ . The additive measurement errors are modelled within the likelihood function.

## 2.1 An example

Consider the case with  $n_d$  range difference measurements,  $n_r$  range measurements, and  $n_p$  planar measurements. Then the measurement vector is

$$y = [d_1 \cdots d_{n_d} \quad r_1 \cdots r_{n_r} \quad a_1 \cdots a_{n_p}]^T$$

and the measurement model

$$h(x) = \begin{bmatrix} \|s_1 - x\| - \|s_0 - x\| \\ \vdots \\ \|s_{n_d} - x\| - \|s_0 - x\| \\ \|s_{n_d+1} - x\| \\ \vdots \\ \|s_{n_d+n_r} - x\| \\ u_1^T x \\ \vdots \\ u_{n_p}^T x \end{bmatrix}.$$

If all the measurement errors have normal distributions, then the likelihood function is

$$L(x) \propto e^{-\frac{1}{2}(h(x)-y)^T \Sigma^{-1}(h(x)-y)}$$

where  $\Sigma$  is the covariance matrix of the measurement errors  $v$ .

## 2.2 Significant domain

The *significant domain*  $S$  is the (simply connected) region in  $\mathbb{R}^d$  in which the solution probability is non-negligible [8]. The numerical solution process can then be constrained inside  $S$  without too much error.

The significant domain can be formed by truncating the measurement likelihoods or prior distribution, or propagating the significant domain of the previous time instant through the (truncated) state model. Truncation must be done carefully to keep the total probability mass outside the significant domain within predefined limits. Additionally, in the local positioning application, the significant domain can further be reduced using, for example, base station maximum ranges, base station sector constraints, altitude constraints, and limiting the user velocity within some reasonable bounds.

### 3 Parallelepiped grid

Model the probability density functions in the Bayesian filtering problem as piecewise constant over a finite collection bounded of *cells*. An evenly spaced  $d$ -dimensional parallelepiped grid can be uniquely described with vectors  $e \in \mathbb{R}^d$ ,  $n \in \mathbb{Z}_+^d$  and a non-singular basis matrix  $E \in \mathbb{R}^{d \times d}$ . The cell centres are then computed from

$$c(i) = e + Ei, \quad 0 \leq i \leq n,$$

where  $i \in \mathbb{Z}_+^d$ . The cell with multi-index  $i$  is defined as

$$G(i) = \left\{ c(i) + E\gamma \mid \gamma \in \left(-\frac{1}{2}, \frac{1}{2}\right]^d \right\}.$$

The area (volume) of each cell is  $|\det E|$ . It is easy to see that the cells are disjoint and cover a parallelepiped region. Figure 1 illustrates a two-dimensional grid with  $n = (2, 4)$ .

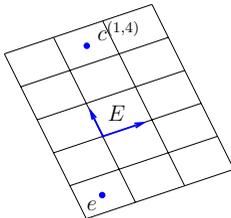


Figure 1: Uniform parallelepiped grid

Let the mean density in a cell be  $\pi(i) \in \mathbb{R}^+$ , so that we can write density function approximations in the form

$$\hat{p}(x) = \sum_{i=0}^n \pi(i) \chi_{G(i)}(x), \quad (5)$$

where  $\chi_{G(i)}$  is the characteristic function of the cell  $G(i)$  [9].

#### 3.1 Grid design

By grid design we mean deciding what area the grid should cover and how large cells to use. When choosing the number of cells to use, one extreme choice is to generate massive amount of very small cells. Then the approximation will be asymptotically accurate even if the  $\pi(i)$  are chosen sub-optimally, for example using just the density value in the cell centre as is done in the classical point-mass filter [10].

Another extreme is to use just a few very large cells. Then it is essential that the mean densities in the cells be computed as accurately as possible. Most of the structure of the pdf is however lost when large parts of the pdf are approximated with constant patches. Fortunately, it usually is not necessary to use grid spacing much denser than the expected positioning accuracy.

In the implementation it is thus possible to try to strike a balance between computation load and accuracy. Optimally, the cell should not be much smaller than the finest features of the posterior distribution. There are some analytic results on how to anticipatively design 1D and 2D grids for the point-mass method [11], but these are yet to be generalised for the multidimensional piecewise constant grids.

### 3.2 Moving grid algorithm

Using the grid approximation, the Bayesian filter can be approximated with the following algorithm. The grid approximation of  $p$  is denoted with  $\hat{p}$ .

**Step 0.** Approximate the initial distribution  $p_{0|0}$  with  $\hat{p}_{0|0}$ . Set  $k = 1$ .

**Step 1.** Prediction step:

$$\hat{p}_{k|k-1}(x) = \int_{\mathbb{R}^d} \phi_{k-1}(x | \xi) \hat{p}_{k-1|k-1}(\xi) d\xi \approx \sum_{i=0}^{n_k} \pi_{k|k-1}(i) \chi_{G_k(i)}(x), \quad (6)$$

where

$$\pi_{k|k-1}(i) = \frac{1}{|\det E_k|} \sum_{j=0}^{n_{k-1}} \pi_{k-1|k-1}(j) \Gamma_k(i|j) \quad (7)$$

and  $\Gamma_k(i|j)$  is the transition probability from  $j$ th cell of  $(k-1)$ th grid to  $i$ th cell of  $k$ th grid.

**Step 2.** Update step:

$$\hat{p}_{k|k}(x) \propto \hat{p}_{k|k-1}(x) L_k(x) \approx \sum_{i=0}^{n_k} \pi_{k|k}(i) \chi_{G_k(i)}(x), \quad (8)$$

where

$$\pi_{k|k}(i) = \pi_{k|k-1}(i) \int_{G_k(i)} L_k(\xi) d\xi. \quad (9)$$

This step requires either the analytic or numerical integration of the likelihood function over each cell.

**Step 3.** Output current mean and variance estimates

$$\begin{aligned} \mu_k &= |\det E_k| \sum_{i=0}^{n_k} \pi_{k|k}(i) c_k(i) \\ \Sigma_k &= |\det E_k| \sum_{i=0}^{n_k} \pi_{k|k}(i) c_k(i) c_k(i)^T - \mu_k \mu_k^T + \frac{|\det E_k|}{12} E_k E_k^T. \end{aligned} \quad (10)$$

**Step 4.** Increase  $k$  and repeat from Step 1.

### 3.3 Cell transition probabilities

In the prediction step (6), we need to compute the new mean density values  $\pi_{k|k-1}(i)$ . The old grid and the new grid are in general different. Denoting the cell volume with  $\alpha_k = |\det E_k|$ , we can compute the mean density in a cell by integrating the predictive pdf (3) over the cell:

$$\pi_{k|k-1}(i) = \frac{1}{\alpha_k} \int_{G_k(i)} p_{k|k-1}(\nu) d\nu = \frac{1}{\alpha_k} \int_{G_k(i)} \left[ \int \phi_{k-1}(\nu | \xi) p_{k-1|k-1}(\xi) d\xi \right] d\nu.$$

Replacing  $p_{k-1|k-1}$  with its grid approximation yields

$$\pi_{k|k-1}(i) \approx \frac{1}{\alpha_k} \sum_{j=0}^{n_{k-1}} \pi_{k-1|k-1}(j) \Gamma_k(i|j), \quad (11)$$

where  $\Gamma_k(i|j)$  stands for the transition probability from  $j$ th cell to  $i$ th cell. Computing this probability efficiently is crucial. Note that it does not depend (directly) on measurements or current state, but only on the state model and the relation of the consecutive grids. As the grid evolves dynamically, the transition probabilities have to be re-computed for each time step.

One implementation possibility is the point-mass filter [10], where we make the approximation  $\Gamma_k(i|j) \approx \phi_k(c_k(i)|c_{k-1}(j))$  where  $c_k(i)$  is the centre point of  $i$ th grid cell. This is fast to compute per cell, but is accurate only for relatively small cells.

To get a more efficient prediction formula (7), we have to assume that the motion model is linear, e.g.  $f(x) \equiv Tx$ , and that the new grid basis is formed by applying the motion model to the old grid basis:  $E_k = TE_{k-1}$ . Then the transition probability between the  $i$ th cell in the old grid and the  $j$ th cell of the new grid depends only on the difference  $i - j$ , and we can write  $\Gamma_k(i|j) = \tau_k(i - j)$ . Then the mean density in the new grid cell (11) becomes

$$\pi_{k|k-1}(i) \approx \frac{1}{\alpha_k} \sum_{j=0}^{n_{k-1}} \pi_{k-1|k-1}(j) \tau_k(i - j),$$

which is fast to compute as a  $d$ -dimensional discrete linear convolution.

Now the transition probability  $\tau_k(i - j)$  is

$$\tau_k(i - j) = \frac{1}{\alpha_{k-1}\alpha_k} \int_{c_k(i)+E_kK} \int_{c_{k-1}(j)+E_{k-1}K} \phi(\nu | \xi) d\xi d\nu,$$

where  $K = (-\frac{1}{2}, \frac{1}{2}]^d$ . This can be simplified to

$$\tau_k(i - j) = \int_K p_{w_k} [E_k(i - j) + e_k - Te_{k-1} + E_k\lambda] d\lambda = \int_K p_{w_k, i-j}(\lambda) d\lambda, \quad (12)$$

where  $p_{w_k, i-j}$  is shorthand for the modified process noise probability density function. Specifically, if  $w_k \sim N(0, Q)$ , then

$$w_{k, i-j} \sim N(j - i - E_k^{-1}(e_k - Te_{k-1}), (E_k^{-1})^T Q E_k^{-1}).$$

The integral (12) then is just multinormal probability in a hyper-box and can be computed numerically [12].

If  $w_k$  is non-Gaussian, the transition probabilities  $\tau_k(i - j)$  can be computed using the cumulative distribution.

### 3.4 Interpolation between grids

Because the consecutive grid bases are connected by  $E_{k+1} = TE_k$ , the cells will eventually become “sheared” along the velocity dimension. In practice, it is best to straighten the grid by interpolating every once in a while. Figure 2 illustrates the interpolation process. In accurate interpolation, the mean density in the new cell is computed by taking intersection with every cell in the old grid and summing the weighted densities. Accurate interpolation is computationally expensive in the general multidimensional case, and is still work in progress. With a large number of small cells, the interpolation method does not have to be very sophisticated, as long as it is asymptotically correct. With a few large cells, on the other hand, it is important to lose as little information as possible in the interpolation.

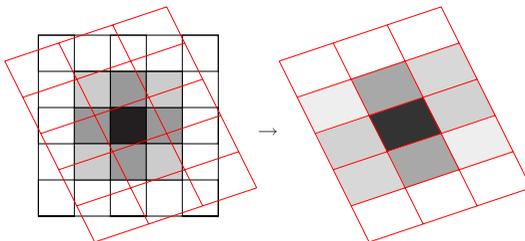


Figure 2: Interpolating from one grid to another

## 4 Testing

The moving grid method was implemented in MATLAB along with a test bench for comparison between other nonlinear filters. Polytope manipulation necessary for managing the significant domains was based on the double description method [13], for which a C++ implementation as well as a MATLAB wrapper [14] are freely available.

The simulation test bench was designed to produce dynamic test data similar to what could be expected in real-world personal positioning scenario. The main difference from the real data is that in the simulation the true track and correct measurement and motion models are available. The testing process consists of first generating a true track of 120 points with one second intervals with a velocity-restricted random walk model, then generating a set of base station along the track with maximum ranges set so that one to three stations can be heard from every point on the track. A GPS constellation is then simulated with an elevation mask and shadowing profile set so that only a couple of satellites are visible at a time. Finally, noisy measurements are generated for each time step from the visible satellite ranges and delta-ranges, base stations ranges and sector information, and optionally also compass and altitude measurements.

Several track and measurement sets were generated with different parameters for the user motion model, measurement sources available, and measurement noises. The testing scenarios range from positioning with just one or two base stations with up to 500 m ranging errors through a hybrid case with a couple of base stations and a two-three satellites to an over-determined satellite-only case. The resulting test set of 622 tracks, hopefully, is representative of the various difficult situations where personal positioning might be used.

The test tracks were run through the moving grid filter, and the mean and covariance of the posterior distribution recorded at each time step. For comparison, the data was also processed with an extended Kalman filter (EKF) [3], and a 2-million point “bootstrap” particle filter, which requires about hundred times more computation but is what we think the closest we can get to the optimal solution without spending months of CPU time. The particle filter solution is used as reference in the testing.

### 4.1 Results

The focus of these preliminary results is on gaining insight into the problem and the behaviour of the grid solver by inspecting the results of some individual runs. Quantitative results would not make much sense at this point because firstly, the data is simulated with arbitrarily chosen parameters so we can expect arbitrary results, and secondly, there is not yet a sound and fair way of comparing different solvers either to each other or to the true/optimal solution.

Figure 3 shows an example run. The ellipses represent the one-sigma approximation of the estimated posterior distributions. The optimal posterior distribution is bi-modal, and the optimal mean estimate

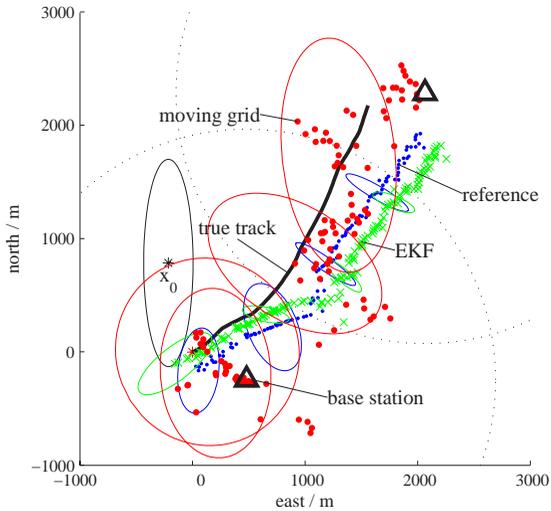


Figure 3: Example run using range measurements from two base stations. The ellipsoids represent the 66% confidence regions.

travels somewhere in between the two. It can be seen how the EKF strayed on the wrong branch of the posterior and gives over-optimistic variance estimates.

The 2D error of the grid filter is compared to the cell radius in Figure 4. The error of the estimated mean from the reference mean, that is taken to be quite close to the optimal posterior mean, is indeed always smaller than the cell radius. Thus, the grid filter works as well as is to be expected. The accuracy can be improved by either using more cells, which results in more computation, or to use a tighter significant domain, in which case there is a risk of leaving too much of the posterior density outside the grid.

## 5 Discussion

Personal positioning often is required although there are only a few measurement sources available that might have large errors with unusual distributions. In these cases, it is essential that the maximum amount of information be extracted from every measurement. One way of achieving this is to approximate the ideal Bayesian filter as accurately as possible.

The moving grid filter presented in this paper is a conservative approach to Bayesian filtering. It may not be able to present the finer features of posterior density function accurately, but it should at least keep track of the shape of the posterior and not drop any its peaks. Preliminary testing of the MATLAB implementation shows that although accurate solution would require huge number of grid cells, the filter can be run with very coarse grids and it succeeds in keeping the magnitude of the error about the same as the cell radius.

This work also underlines the question of how to evaluate the performance of nonlinear filters. Should we compare the absolute error between the posterior mean and the true track? When the posterior distribution has two distinct peaks, the posterior mean is between the peaks and can be far away from the

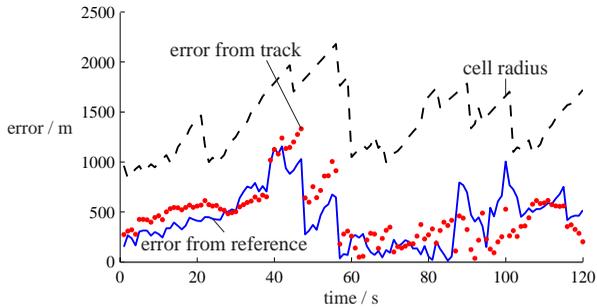


Figure 4: The 2D error of the moving grid filter mean estimate compared to the cell radius

true track. Then discarding one of the peaks might lead to seemingly better performance even though the posterior distribution is clearly wrong. One approach would be to compare the maximum posterior estimate to the true position, but then there would be the problem of what to do when the posterior approximation has several peaks. Ideally, we would like to compare the approximated posterior distribution to the true posterior distribution. The problem with this is that while the true track might be known in the simulation, the true posterior implied by the measurements is not.

In this work, we resorted to comparing the filtered mean with the true track and with the means of the classical EKF and that of a 2-million point particle filter. It would have been informative to compare the actual posterior distributions of the different filters, but as of yet no feasible numerical method of doing so is not known. This is a topic for future research.

## Acknowledgments

This study was funded by Nokia Corporation. Both authors were supported in part by the Nokia Foundation. The particle filter we used for reference was implemented by Duane Petrovich.

## References

- [1] J. Syrjärinne, *Studies of modern techniques for personal positioning*, Dissertation, Tampere University of Technology, Tampere, 2001.
- [2] A. Kotanen, M. Hännikäinen, H. Leppäkoski, and T. D. Hämäläinen, "Experiments on local positioning with Bluetooth," in *Proceedings of the International Conference on Information Technology: Computers and Communications (ITCC'03)*, 2003, pp. 297–303.
- [3] S. Ali-Löytty, N. Sirola, and R. Piché, "Consistency of three Kalman filter extensions in hybrid navigation," in *Proceedings of the European Navigation Conference GNSS 2005*, Munchen, July 19-22, 2005.
- [4] D. Fox, J. Hightower, H. Kautz, L. Liao, and D. J. Patterson, "Bayesian techniques for location estimation," in *Proceedings of Workshop on Location-aware Computing, part of UBIComp Conference*, Seattle, October 2003, pp. 16–18.
- [5] T. Roos, P. Myllymäki, and H. Tirri, "A statistical modeling approach to location estimation," *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, pp. 59–69, 2002.

- [6] K. Heine, "Unified framework for sampling/importance resampling algorithms," in *Proceedings of Fusion 2005*, Philadelphia, July 25-29, 2005.
- [7] A. Farina, D. Benvenuti, and B. Ristic, "A comparative study on the Benes filtering problem," *Signal Processing*, vol. 82, pp. 133–147, 2002.
- [8] S. Challa and Y. Bar-Shalom, "Nonlinear filter design using Fokker-Planck-Kolmogorov probability density evolutions," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 1, pp. 309–315, 2000.
- [9] S. C. Kramer and H. W. Sorenson, Recursive Bayesian estimation using piece-wise constant approximations, *Automatica*, vol. 24, no. 6, pp. 789–801, 1988.
- [10] R. S. Bucy and K. D. Senne, "Digital synthesis of non-linear filters," *Automatica*, vol. 7, no. 3, pp. 287–298, 1971.
- [11] M. Šimandl, J. Královec, and T. Söderström, "Anticipative grid design in point-mass approach to nonlinear state estimation," *IEEE Transactions on Automatic Control*, vol. 47, no. 4, pp. 699–702, April 2002.
- [12] A. Genz, "Numerical computation of multivariate normal probabilities," *Journal of Computational and Graphical Statistics*, vol. 1, no. 1, pp. 141–149, 1992.
- [13] K. Fukuda and A. Prodon, "Double description method revisited," in M. Deza, R. Euler, and I. Manoussakis (Eds.), *Combinatorics and Computer Science*, volume 1120 of *Lecture Notes in Computer Science*, pp. 91–111, Springer-Verlag, 1996.
- [14] F. Torrisi and M. Baotic, CDDMEX - Matlab interface for the CDD solver, web page, August 2005. Available at <http://control.ee.ethz.ch/~hybrid/cdd.php>