

# Notes

## Related Work:

Greytak [?] provides a detailed treatment of how to evaluate the probability of success for a point robot under motion uncertainty by considering truncated Gaussians but does not consider sensing uncertainty or a feedback law for dealing with the uncertainty. Toussaint [?] truncates the Gaussian belief to compute a revised distribution over the state in the context of approximate inference control (which provides a solution to the stochastic optimal control problem) but does not explicitly compute the probability of success. Erez and Smart [?] use the approach suggested in [?] to approximate the Gaussian belief (represented as a Gaussian mixture) along one active constraint and solve the stochastic optimal control problem using differential dynamic programming (DDP) in belief space. Their method does not easily generalize to multiple constraints. Vitus and Tomlin [?] use Boole's inequality to compute a conservative bound for the probability of success to solve a chance constrained optimal control problem. They do not truncate the beliefs while processing constraints, thereby obtaining a very conservative estimate of the actual probability of success.

There is also a considerable body of work on Kalman filtering with state constraints. [?] outlines several methods to perform state estimation under (linear and nonlinear) equality and inequality state constraints, one of which is to truncate Gaussian densities along the specified constraints.

## Truncated Gaussians:

Given a  $n$ -dimensional Gaussian belief  $\mathcal{N}[\mathbf{x}|\hat{\mathbf{x}}, \Sigma]$  and a (locally-)convex feasible region  $\mathcal{X}_{free}^C$  bound by  $m$  hyperplanes  $\{\mathbf{a}_i^T \mathbf{x} - b_i = 0\}$ :  $\mathcal{X}_{free}^C \equiv \bigcap_{i=1}^m \{\mathbf{a}_i^T \mathbf{x} > b_i\}$ , we want to estimate the truncated Gaussian  $\mathcal{N}[\mathbf{x}|\hat{\mathbf{x}}, \Sigma]$  that best approximates the Gaussian distribution of the samples drawn from  $\mathcal{N}[\mathbf{x}|\hat{\mathbf{x}}, \Sigma]$  that lie in  $\mathcal{X}_{free}^C$ .

We propose to construct the truncated Gaussian  $\mathcal{N}[\mathbf{x}|\hat{\mathbf{x}}, \Sigma]$  by sequentially truncating the original Gaussian  $\mathcal{N}[\mathbf{x}|\hat{\mathbf{x}}, \Sigma]$  along each of the  $m$  half-space constraints:  $\{\mathbf{a}_i^T \mathbf{x} > b_i\}, i = 1 \dots, m$ , where  $\mathbf{a}_i$  is the  $i^{\text{th}}$  hyperplane normal and  $b_i$  is the offset. Let  $\mathcal{N}[\mathbf{x}|\hat{\mathbf{x}}_{i-1}, \Sigma_{i-1}]$  be the Gaussian obtained after sequentially truncating  $\mathcal{N}[\mathbf{x}|\hat{\mathbf{x}}, \Sigma]$  along  $(i-1)$  constraints.

Following the derivation of [?], we process the  $i^{\text{th}}$  constraint by first transforming the problem such that the Gaussian  $\mathcal{N}[\mathbf{x}|\hat{\mathbf{x}}_{i-1}, \Sigma_{i-1}]$  becomes a standard Gaussian and the constraint is aligned along a reference unit vector  $\mathbf{e} = (1, 0, \dots, 0)$ . If  $\Sigma_{i-1} = C^T C$  be the Cholesky decomposition ( $\Sigma_{i-1}^{-1} = C^{-1} C^{-T}$ ), we can express the Gaussian  $\mathcal{N}[\mathbf{x}|\hat{\mathbf{x}}_{i-1}, \Sigma_{i-1}]$  in the coordinate system of  $\mathbf{y} = C^{-T}(\mathbf{x} - \hat{\mathbf{x}}_{i-1})$  as:

$$\mathcal{N}[\mathbf{x}|\hat{\mathbf{x}}_{i-1}, \Sigma_{i-1}] \propto \exp\left\{-\frac{1}{2}(\mathbf{x} - \hat{\mathbf{x}}_{i-1})^T \Sigma^{-1}(\mathbf{x} - \hat{\mathbf{x}}_{i-1})\right\} \longrightarrow \exp\left\{-\frac{1}{2}\mathbf{y}^T \mathbf{y}\right\} \equiv \mathcal{N}[\mathbf{y}|\mathbf{0}, I]$$

where  $\mathcal{N}[\mathbf{y}|\mathbf{0}, I]$  is a Gaussian with zero mean and unit variance. By substituting  $\mathbf{x} = (C^T \mathbf{y} + \hat{\mathbf{x}}_{i-1})$ , the constraint  $\{\mathbf{a}_i^T \mathbf{x} > b_i\}$  is transformed to  $\{\mathbf{u}_i^T \mathbf{y} > v_i\}$ , where  $\mathbf{u}_i = C\mathbf{a}_i/|C\mathbf{a}_i|$  is a normalized vector and  $v_i = (b_i - \mathbf{a}_i^T \hat{\mathbf{x}}_{i-1})/|C\mathbf{a}_i|$ .

We compute a rotation matrix  $R$  that rotates the reference unit vector  $\mathbf{e} = (1, 0, \dots, 0)$  onto the normalized vector  $\mathbf{u}_i$  i.e.  $\mathbf{u}_i = R\mathbf{e}$ . The Gaussian  $\mathcal{N}[\mathbf{y}|\mathbf{0}, I]$  and the constraint are transformed in a rotated coordinate system  $\mathbf{y}' = R^{-1}\mathbf{y}$  as  $\mathcal{N}[\mathbf{y}'|\mathbf{0}, I]$  and  $\{(R^{-1}\mathbf{u}_i)^T \mathbf{y}' > v_i\} \equiv \{\mathbf{e}^T \mathbf{y}' > v_i\}$  respectively. The problem

is now transformed to a simplified 1D form where the standard Gaussian distribution  $\mathcal{N}[\mathbf{y}'|\mathbf{0}, I]$  is truncated along the first axis in the  $\mathbf{y}'$  coordinate system.

The mean and variance of a standard Gaussian distribution  $\mathcal{N}[v|0, 1]$  truncated along a linear constraint  $v \geq v_i$  are given by  $\mu = (e^{-v_i^2/2}/n)$  and  $\sigma^2 = (1 + v_i\mu - \mu^2)$  respectively, where  $n = \sqrt{\pi/2}(1 - \text{erf}(v_i/\sqrt{2}))$  [?]. Given the mean  $\mu$  and variance  $\sigma^2$  of the  $v_i$ -truncated standard Gaussian, the truncated standard Gaussian in the  $\mathbf{y}'$  coordinate system is obtained as  $\mathcal{N}[\mathbf{y}'|(\mu, 0, \dots, 0)^T, \text{diag}(\sigma^2, 1, \dots, 1)]$ . The mean and variance of the  $i^{\text{th}}$  truncated Gaussian  $\mathcal{N}[\mathbf{x}|\hat{\mathbf{x}}_i, \Sigma_i]$  are then obtained by applying the reverse transform  $\mathbf{x} = C^T R \mathbf{y}' + \hat{\mathbf{x}}_{i-1}$ :

$$\begin{aligned}\hat{\mathbf{x}}_i &= C^T R \cdot (\mu, 0, \dots, 0)^T + \hat{\mathbf{x}}_{i-1} \\ \Sigma_i &= C^T R \cdot \text{diag}(\sigma^2, 1, \dots, 1) \cdot R^T C\end{aligned}$$

We obtain the desired truncated Gaussian  $\mathcal{N}[\mathbf{x}|\hat{\mathbf{x}}, \underline{\Sigma}]$  by sequentially processing the  $m$  half-space constraints i.e.  $\hat{\mathbf{x}} = \hat{\mathbf{x}}_m$  and  $\underline{\Sigma} = \Sigma_m$ . Our experiments indicate that the order in which the constraints are processed is important but the differences in the obtained means and variances are minimal when compared to the truncated Gaussians computed using dense sampling (Fig. ??).

### Analytical Computation of Probability of Success:

Given a  $n$ -dimensional Gaussian belief  $\mathcal{N}[\mathbf{x}|\hat{\mathbf{x}}, \Sigma]$  and a (locally-)convex feasible region  $\mathcal{X}_{free}^C$  bound by  $m$  hyperplanes  $\{\mathbf{a}_i^T \mathbf{x} - b_i = 0\}$ :  $\mathcal{X}_{free}^C \equiv \bigcap_{i=1}^m \{\mathbf{a}_i^T \mathbf{x} > b_i\}$ , we want to estimate the probability of success:  $p(\mathbf{x} \in \mathcal{X}_{free}^C), \mathbf{x} \sim \mathcal{N}[\mathbf{x}|\hat{\mathbf{x}}, \Sigma]$ . Exact computation of this probability requires the integration of a multivariate Gaussian distribution over the feasible region, which does not have an analytic solution. A naïve solution involves densely sampling the Gaussian  $\mathcal{N}[\mathbf{x}|\hat{\mathbf{x}}, \Sigma]$  and counting the number of successful samples that lie in  $\mathcal{X}_{free}^C$  to estimate the probability. This strategy is computationally expensive. We outline a method below to estimate this probability analytically.

**Method 1:** Boole's inequality states that for a countable set of events  $E_1, E_2, \dots$ , the probability that at least one of the events happens is no larger than the sum of the individual probabilities i.e.  $p(\bigcup E_i) \leq \sum p(E_i)$ . By using Boole's inequality, the probability  $p(\mathbf{x} \in \mathcal{X}_{free}^C)$  can be conservatively estimated as [?]:

$$\begin{aligned}p(\mathbf{x} \in \mathcal{X}_{free}^C) &= 1 - p(\mathbf{x} \notin \mathcal{X}_{free}^C) \\ &\geq 1 - \sum_{i=1}^m p(\mathbf{a}_i^T \mathbf{x} \leq b_i)\end{aligned}$$

The probability  $p(\mathbf{a}_i^T \mathbf{x} \leq b_i), \mathbf{x} \sim \mathcal{N}[\mathbf{x}|\hat{\mathbf{x}}, \Sigma]$  along the  $i^{\text{th}}$  half-space constraint is evaluated as:

$$\begin{aligned}p(\mathbf{a}_i^T \mathbf{x} \leq b_i) &= \frac{1}{\sqrt{2\pi \mathbf{a}_i^T \Sigma \mathbf{a}_i}} \int_{-\infty}^{b_i} \exp\left(-\frac{(y - \mathbf{a}_i^T \hat{\mathbf{x}})^2}{2\mathbf{a}_i^T \Sigma \mathbf{a}_i}\right) dy, \quad \text{where } y = \mathbf{a}_i^T \mathbf{x} \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\frac{b_i - \mathbf{a}_i^T \hat{\mathbf{x}}}{\sqrt{\mathbf{a}_i^T \Sigma \mathbf{a}_i}}} \exp\left(-\frac{z^2}{2}\right) dz \\ &= \text{cdf}\left(\frac{b_i - \mathbf{a}_i^T \hat{\mathbf{x}}}{\sqrt{\mathbf{a}_i^T \Sigma \mathbf{a}_i}}\right)\end{aligned}$$

where cdf is the Gaussian cumulative distribution function and is related to the error function erf as  $\text{cdf}(z) = \frac{1}{2}(1 + \text{erf}(\frac{z}{\sqrt{2}}))$ , which in turn can be efficiently evaluated using a series approximation. The probability estimate computed using this method can be very conservative (Figs. ?? and ??).

**Method 2:** Another approach is to sequentially truncate the Gaussian  $\mathcal{N}[\mathbf{x}|\hat{\mathbf{x}}, \Sigma]$  along each of the  $m$  half-space constraints  $\{\mathbf{a}_i^T \mathbf{x} > b_i\}$  and compute the probability  $p(\mathbf{x} \in \mathcal{X}_{free}^C)$  as:

$$\begin{aligned} p(\mathbf{x} \in \mathcal{X}_{free}^C) &= 1 - p(\mathbf{x} \notin \mathcal{X}_{free}^C) \\ &\approx \prod_{i=1}^m (1 - p(\mathbf{a}_i^T \mathbf{x} \leq b_i)) \end{aligned}$$

where  $p(\mathbf{a}_i^T \mathbf{x} \leq b_i)$  is computed at each step using the truncated Gaussian approximation  $\mathcal{N}[\mathbf{x}|\hat{\mathbf{x}}_i, \Sigma_i]$  obtained after processing the  $i^{\text{th}}$  constraint, as:

$$p(\mathbf{a}_i^T \mathbf{x} \leq b_i) = \text{cdf}\left(\frac{b_i - \mathbf{a}_i^T \hat{\mathbf{x}}_i}{\sqrt{\mathbf{a}_i^T \Sigma_i \mathbf{a}_i}}\right)$$

Since this method estimates the probability of success based on successive truncated Gaussian approximations, the order in which the truncated Gaussians are computed is important. In our experiments, we process the constraints in a random order and our experiments indicate that the probability of success computed using this method provides a good, though conservative, estimate of the actual probability (verified using dense sampling) and performs better than the first method (Figs. ?? and ??). We still need to verify if the estimate is always conservative.

## To Do:

- **Local convexification:** We could follow the construction outlined in [?] to automatically decompose the workspace into locally-convex regions of free space using BSP trees. This approach generalizes to both 2D and 3D workspaces. Should the convex decomposition be static (determined in a pre-processing step) or should it be computed on the fly?
- **Propagate truncated beliefs:** We need to propagate the truncated beliefs at each time-step according to the dynamics and observation models while accounting for feedback (similar to LQG-MP). This could also be incorporated in an ILQG framework for local optimization of trajectories in belief space. How do we handle the fact that the mean of the posterior distribution deviates from the path and how can the probability of success be computed a priori to execution?
- **Order of constraints:** The method seems to depend on the order in which the constraints are processed. Is there a way to achieve order independence?
- **Sampling time:** The probability of success computation also depends on the sampling time duration. Since we are dealing with a discrete dynamics and observation model, this is not an issue but would be if we are dealing with continuous process models.
- **Non-point robots:** How can we generalize this method to handle non-point robots? Instead of considering a spherical cover for the non-point robot (oversimplified), maybe we could consider a tight fitting ellipsoid cover for the robot and use Gaussian convolution operations to obtain a better estimate of the probability of success?

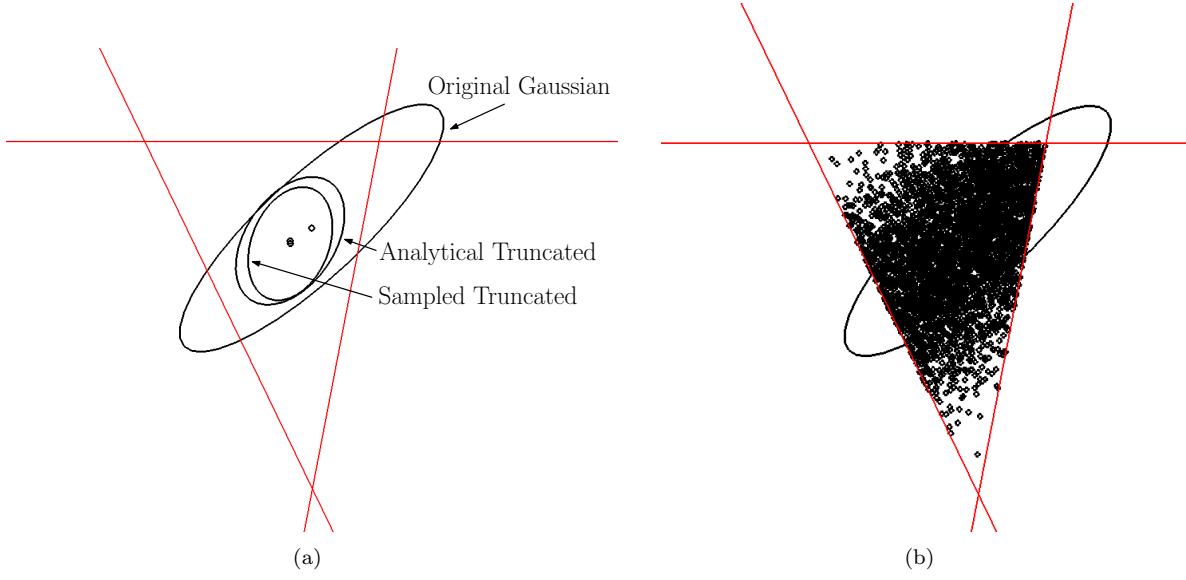


Figure 1: 2D Gaussian truncated by 3 half-space constraints. (a) The analytically computed truncated Gaussian is a good approximation of the sampled truncated Gaussian computed using 10000 samples (shown in (b)). The probability of success using (i) Boole's inequality: 0.182, (ii) Method 2: 0.341, and (iii) Sampling: 0.369. The result obtained using Boole's inequality is overly conservative.

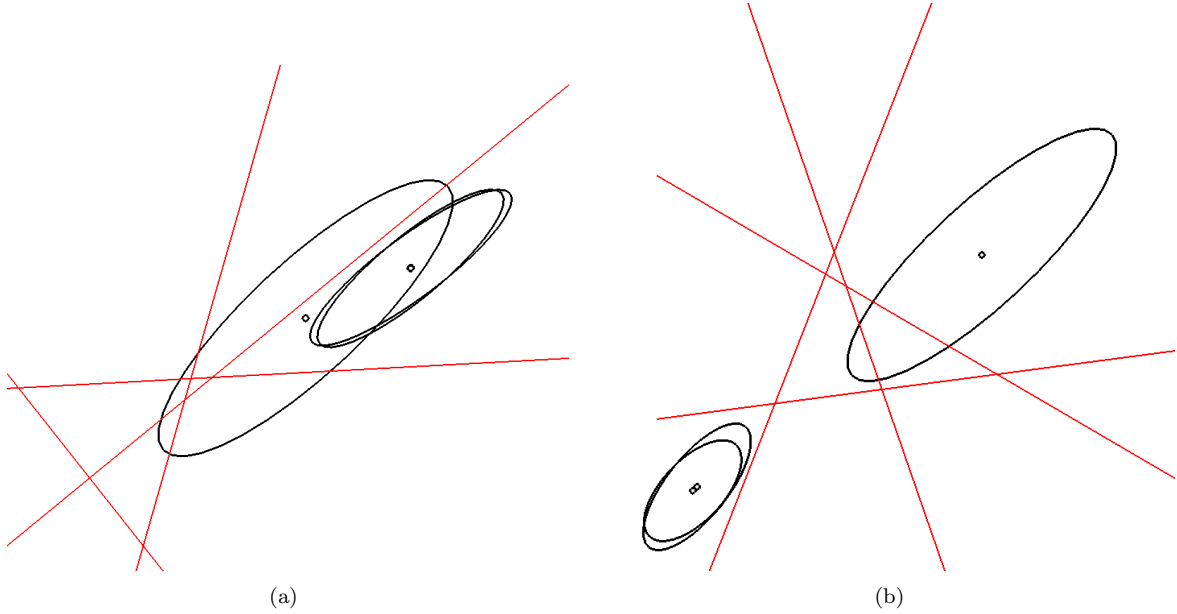


Figure 2: 2D Gaussian truncated by 4 randomly generated half-space constraints. (a) The probability of success using (i) Boole's inequality: 0(-0.03), (ii) Method 2: 0.326, and (iii) Sampling: 0.33. (b) The probability of success using (i) Boole's inequality: 0(-2.3), (ii) Method 2: 0.0339, and (iii) Sampling: 0.0341. The result obtained using Boole's inequality predicts a zero probability of success, which is not the case.

## References

- [1] J. Comba, C. Silva. Automatic Convexification of Space Using BSP trees. *Unpublished manuscript*, 2003. <http://www.sci.utah.edu/~csilva/papers/bsp-fill.pdf>.
- [2] T. Erez, W. D. Smart. A Scalable Method for Solving High-Dimensional Continuous POMDPs Using Local Approximation. *Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2010. <http://research.engineering.wustl.edu/~etom/uai10.pdf>
- [3] M. Greytak. Integrated Motion Planning and Model Learning for Mobile Robots with Application to Marine Vehicles. *PhD Thesis*. MIT. 2009. [http://web.mit.edu/hovergroup/pub/Thesis\\_mgreytak.pdf](http://web.mit.edu/hovergroup/pub/Thesis_mgreytak.pdf)
- [4] D. Simon. Kalman Filtering with State Constraints: A Survey of Linear and Nonlinear Algorithms. *IET Control Theory and Applications*, 2009. <http://academic.csuohio.edu/simond/ConstrKF/ConstrKF.pdf>
- [5] M. Toussaint. Technical Note: Computing moments of a truncated Gaussian for EP in high dimensions. 2009. <http://userpage.fu-berlin.de/~mtoussai/notes/truncatedGaussian.pdf>.
- [6] M. Toussaint. Pros and Cons of truncated Gaussian EP in the context of Approximate Inference Control. *NIPS Workshop on Probabilistic Approaches for Robotics and Control*, 2009. <http://userpage.fu-berlin.de/mtoussai/publications/09-toussaint-NIPSworkshop.pdf>.
- [7] M. P. Vitus, C. J. Tomlin. Closed-loop Belief Space Planning for Linear, Gaussian Systems. *IEEE Int. Conf. on Robotics and Automation*, 2011. [http://www.stanford.edu/~vitus/pubs/VT\\_IFAC\\_11.pdf](http://www.stanford.edu/~vitus/pubs/VT_IFAC_11.pdf)