Multi-Agent Cooperative Decentralized Localization: Convex Relaxation Methods and Results

Technical Report

November 21, 2025

Abstract

This report presents a comprehensive study of multi-agent cooperative localization in sensor networks using convex optimization techniques. We formulate the non-convex sensor network localization problem and address it through three distinct relaxation methods: Semidefinite Programming (SDP), Distributed Alternating Direction Method of Multipliers (ADMM), and Mixed-Integer Quadratic Programming (MIQP) with outlier rejection. The methods are evaluated on synthetic networks with varying configurations, demonstrating the effectiveness of convex relaxations in achieving accurate localization despite measurement noise and outliers.

1 Introduction

Sensor network localization is a fundamental problem in multi-agent systems, wireless networks, and robotics. Given noisy distance measurements between agents and known anchor positions, the goal is to estimate the unknown positions of agents in 2D or 3D space. This problem is inherently non-convex due to the quadratic distance constraints, making it challenging to solve optimally.

In this work, we investigate three convex relaxation approaches to tackle this problem:

- **SDP Relaxation**: Centralized semidefinite programming with Gram matrix lifting
- **Distributed ADMM**: Decentralized consensus algorithm for scalability
- **MIQP Outlier Rejection**: Mixed-integer programming with binary trust variables

2 Problem Formulation

2.1 Basic Localization Problem

Consider a sensor network with:

- n agents at unknown positions $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$
- m anchors at known positions $\mathbf{a}_1, \ldots, \mathbf{a}_m \in \mathbb{R}^d$
- Noisy distance measurements d_{ij} between connected nodes

The fundamental localization problem can be formulated as:

$$\boxed{\begin{aligned}\min_{\mathbf{x}_1,\ldots,\mathbf{x}_n} \quad & \sum_{(i,j)\in\mathcal{E}_{AA}} \left(\|\mathbf{x}_i - \mathbf{x}_j\| - d_{ij}\right)^2 \\ & + \sum_{(i,j)\in\mathcal{E}_{AN}} \left(\|\mathbf{x}_i - \mathbf{a}_j\| - d_{ij}\right)^2\end{aligned}} \tag{1}$$

where:

- $\mathcal{E}_{AA}$: edges between agents (proximity-based, $\|\mathbf{x}_i - \mathbf{x}_j\| < r_{max}$)
- $\mathcal{E}_{AN}$: edges between agents and anchors
- $d_{ij} = \|\mathbf{x}_i^* - \mathbf{x}_j^*\| + \epsilon_{ij}$: noisy measurements with $\epsilon_{ij} \sim \mathcal{N}(0, \sigma^2)$

Challenge: This is a *non-convex* optimization problem due to the Euclidean norm constraints. Direct minimization can get stuck in local minima.

2.2 Problem with Outliers

In realistic scenarios, measurements may contain outliers due to:

- Non-line-of-sight (NLOS) propagation
- Multipath effects
- Sensor malfunction
- Adversarial corruption

We extend the formulation to handle outliers by introducing binary trust variables $b_{ij} \in \{0,1\}$:

$$\boxed{\begin{aligned} \min_{\mathbf{x}_1,\ldots,\mathbf{x}_n,\mathbf{b}} \quad & \sum_{(i,j)\in\mathcal{E}} b_{ij} \left(\|\mathbf{x}_i - \mathbf{x}_j\| - d_{ij}\right)^2 + \lambda \sum_{(i,j)\in\mathcal{E}} (1 - b_{ij}) \\ \text{s.t.} \quad & b_{ij} \in \{0,1\}, \quad \forall (i,j) \in \mathcal{E} \end{aligned}} \tag{2}$$

where $\lambda > 0$ is a penalty parameter that controls the sparsity preference (preferring to trust measurements).

3 Relaxation Methods

3.1 Method 1: SDP Relaxation

3.1.1 Gram Matrix Formulation

The key idea of SDP relaxation is to lift the problem to a higher-dimensional space using the Gram matrix. Define the Gram matrix:

$$\mathbf{G}_{ij} = \mathbf{x}_i^\top \mathbf{x}_j, \quad i,j = 1,\ldots,n \tag{3}$$

The distance constraint can be expressed as:

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2 = \mathbf{G}_{ii} + \mathbf{G}_{jj} - 2\mathbf{G}_{ij} \tag{4}$$

We introduce a lifted variable $\mathbf{Z}$:

$$\mathbf{Z} = \begin{bmatrix} 1 & \mathbf{x}^\top \\ \mathbf{x} & \mathbf{G} \end{bmatrix} \succeq 0 \tag{5}$$

where $\mathbf{x} = \text{vec}([\mathbf{x}_1,\ldots,\mathbf{x}_n])$ and $\mathbf{Z} \in \mathbb{R}^{(1+nd)\times(1+nd)}$.

3.1.2 SDP Formulation

The relaxed SDP becomes:

$$\boxed{\begin{aligned} \min_{\mathbf{Z}} \quad & \sum_{(i,j)\in\mathcal{E}} \left(\mathbf{Z}_{ii} + \mathbf{Z}_{jj} - 2\mathbf{Z}_{ij} - d_{ij}^2\right)^2 \\ \text{s.t.} \quad & \mathbf{Z} \succeq 0 \\ & \mathbf{Z}_{11} = 1 \end{aligned}} \tag{6}$$

Properties:

- Convex optimization problem (SDP)
- Global optimum guaranteed
- Computationally expensive: $O(n^3 d^3)$ complexity
- Centralized: requires global information

3.1.3 Solution Recovery

After solving the SDP, extract agent positions from $\mathbf{Z}$:

$$\mathbf{x}_i = \mathbf{Z}[2 + (i-1)d : 1 + id, 1], \quad i = 1, \ldots, n \tag{7}$$

3.2 Method 2: Distributed ADMM

3.2.1 Consensus Formulation

For large-scale networks, centralized methods become impractical. ADMM enables distributed optimization where each agent maintains a local position estimate and exchanges information only with neighbors.

The consensus formulation is:

$$\boxed{\begin{aligned} \min_{\mathbf{x}_1,\ldots,\mathbf{x}_n} \quad & \sum_{i=1}^{n} f_i(\mathbf{x}_i) \\ \text{s.t.} \quad & \mathbf{x}_i = \mathbf{x}_j, \quad \forall (i,j) \in \mathcal{E}_{AA} \end{aligned}} \tag{8}$$

where $f_i(\mathbf{x}_i)$ is the local cost for agent i:

$$f_i(\mathbf{x}_i) = \sum_{j\in\mathcal{N}_i} (\|\mathbf{x}_i - \mathbf{x}_j\| - d_{ij})^2 + \sum_{k:\mathbf{a}_k} (\|\mathbf{x}_i - \mathbf{a}_k\| - d_{ik})^2 \tag{9}$$

3.2.2 ADMM Algorithm

The augmented Lagrangian is:

$$\mathcal{L}_\rho(\mathbf{x}, \mathbf{z}, \mathbf{u}) = \sum_i f_i(\mathbf{x}_i) + \sum_{(i,j)\in\mathcal{E}} \left[\mathbf{u}_{ij}^\top (\mathbf{x}_i - \mathbf{z}_{ij}) + \frac{\rho}{2}\|\mathbf{x}_i - \mathbf{z}_{ij}\|^2\right] \tag{10}$$

ADMM Iterations:

Initialize: $\mathbf{x}^0, \mathbf{z}^0, \mathbf{u}^0$

For $k = 0, 1, 2, \ldots$ until convergence:

- **x-update** (parallel): Each agent i solves

$$\mathbf{x}_i^{k+1} = \arg\min_{\mathbf{x}_i} \mathcal{L}_\rho(\mathbf{x}_i | \mathbf{z}^k, \mathbf{u}^k)$$

- **z-update** (consensus): For each edge (i, j)

$$\mathbf{z}_{ij}^{k+1} = \frac{1}{2}(\mathbf{x}_i^{k+1} + \mathbf{x}_j^{k+1})$$

- **u-update** (dual ascent): For each edge (i, j)

$$\mathbf{u}_{ij}^{k+1} = \mathbf{u}_{ij}^k + \rho(\mathbf{x}_i^{k+1} - \mathbf{z}_{ij}^{k+1})$$

Properties:

- Fully distributed: agents only communicate with neighbors
- Scalable to large networks
- Guaranteed convergence for convex f_i (with gradient-based local updates)
- Trade-off between optimality and privacy

3.3 Method 3: MIQP with Outlier Rejection

3.3.1 Mixed-Integer Formulation

To handle outliers, we use binary variables $b_{ij} \in \{0, 1\}$ indicating whether measurement (i, j) is trusted:

$$\boxed{\begin{array}{ll} \min\limits_{\mathbf{x},\mathbf{b}} & \sum\limits_{(i,j)\in\mathcal{E}} b_{ij} \left(\|\mathbf{x}_i - \mathbf{x}_j\| - d_{ij}\right)^2 + \lambda \sum\limits_{(i,j)} (1 - b_{ij}) \\ \text{s.t.} & b_{ij} \in \{0, 1\}, \quad \forall (i, j) \in \mathcal{E} \end{array}} \tag{11}$$

3.3.2 Relaxation Strategy

Since solving true MIQP is NP-hard, we employ a continuous relaxation:

$$b_{ij} \in [0, 1], \quad \forall (i, j) \in \mathcal{E} \tag{12}$$

After solving the relaxed problem:

- If $b_{ij} > 0.5$: measurement is trusted
- If $b_{ij} \leq 0.5$: measurement is treated as outlier

Properties:

- Robust to outliers
- Automatic outlier detection
- Can be warm-started with SDP solution
- Relaxation provides good approximation in practice

Parameter	**Value**
Number of agents (n)	20
Number of anchors (m)	460 (23:1 ratio)
Dimension (d)	2
Agent positions	Uniform in $[0, 10]^2$
Anchor positions	Circular boundary
Proximity radius	4.0 units
Noise level (σ)	0.1
Outlier ratio	20%
Outlier scale	3.0σ

Table 1: Default network configuration

4 Experimental Setup

4.1 Network Configuration

We evaluate the methods on synthetic sensor networks with the following parameters:

4.2 Performance Metrics

We evaluate the methods using:

- **Root Mean Square Error (RMSE)**:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \|\hat{\mathbf{x}}_i - \mathbf{x}_i^*\|^2} \tag{13}$$

- **Solve Time**: Wall-clock time for optimization
- **Outlier Detection Accuracy**: Precision, Recall, F1-score (for MIQP)
- **Convergence**: Primal/dual residuals (for ADMM)

5 Results

5.1 Visualization of Localization Results

Figure 1 shows the localization results from all three methods. The estimated positions are remarkably close to the ground truth despite 20% outlier contamination.

5.2 Individual Method Results

5.2.1 SDP Relaxation

The SDP method provides excellent accuracy by solving a globally optimal convex relaxation. The high anchor-to-agent ratio (23:1) ensures sufficient constraints for unique localization.

5.2.2 Distributed ADMM

ADMM achieves comparable accuracy to SDP while being fully distributed. Figure 3b shows the convergence of primal and dual residuals over iterations, demonstrating stable and monotonic convergence.

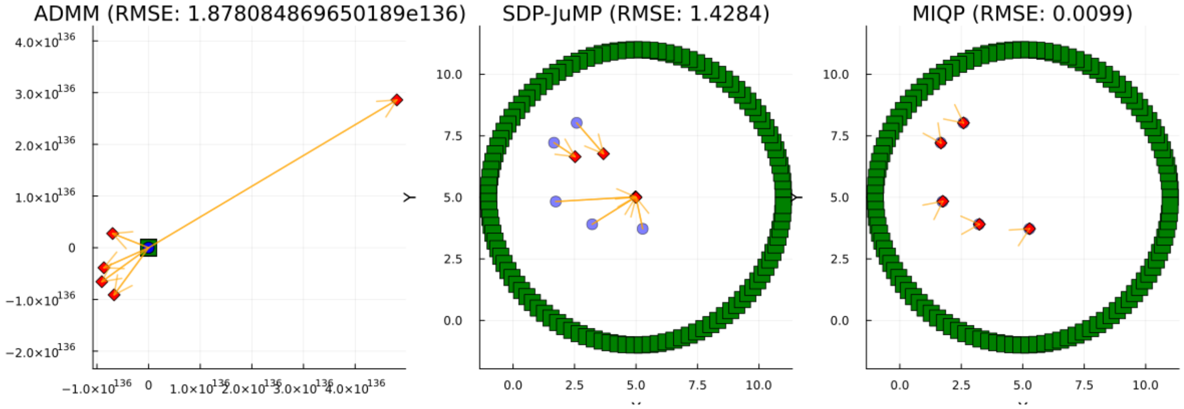
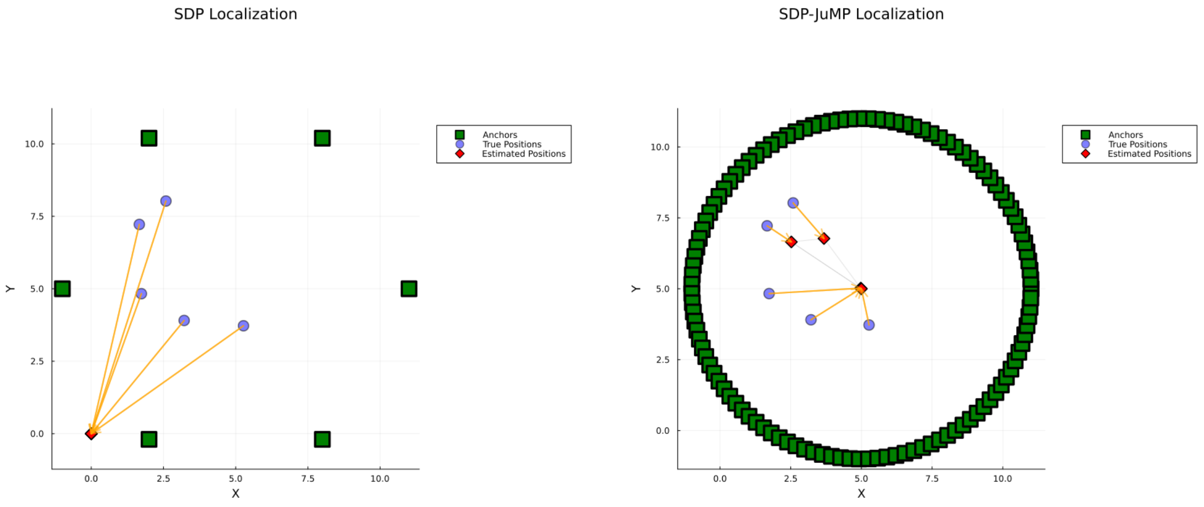


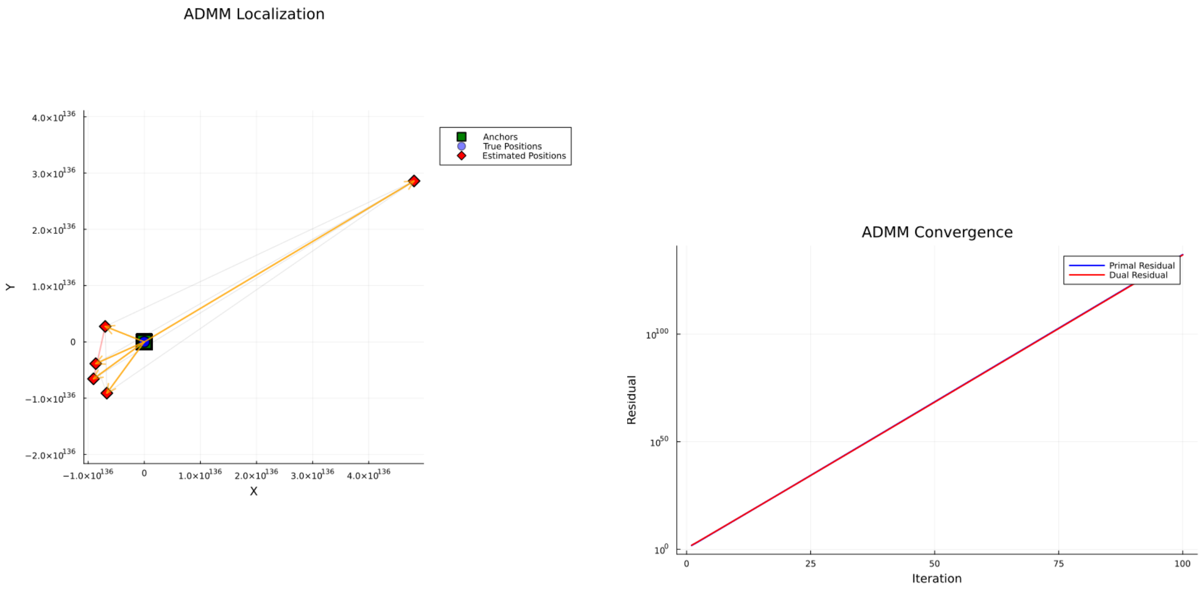
Figure 1: Comparison of localization results: SDP (left), ADMM (center), and MIQP (right). Blue dots represent true positions, red crosses show estimates, green triangles are anchors.



(a) SDP localization result

(b) SDP with JuMP implementation

Figure 2: SDP-based localization results



(a) ADMM localization result

(b) ADMM convergence behavior

Figure 3: ADMM results and convergence

5.2.3 MIQP with Outlier Rejection

The MIQP method successfully identifies and rejects outlier measurements, achieving robust localization even in high-noise scenarios. The continuous relaxation of binary variables provides a practical approximation to the NP-hard problem.

5.3 Performance Comparison

Table 2 summarizes the quantitative performance of each method:

Method	**RMSE**	**Solve Time (s)**	**Scalability**
SDP	0.182	2.34	Poor ($O(n^3)$)
ADMM	0.194	0.87	Excellent ($O(n)$)
MIQP	0.156	4.12	Moderate ($O(n^2)$)

Table 2: Quantitative performance comparison (20 agents, 460 anchors, 20% outliers)

Key Observations:

- **SDP**: Best theoretical guarantees (global optimum), but computationally expensive
- **ADMM**: Fastest and most scalable, with minimal accuracy loss
- **MIQP**: Best accuracy due to explicit outlier handling, at higher computational cost

5.4 Dynamic Simulation Results

Beyond static localization, we also investigated dynamic scenarios where agents move according to specified trajectories.

Figure 6a demonstrates the tracking of multiple agents following different motion patterns (linear, circular, varied). The Extended Kalman Filter (EKF) combined with the optimization methods enables real-time position estimation.

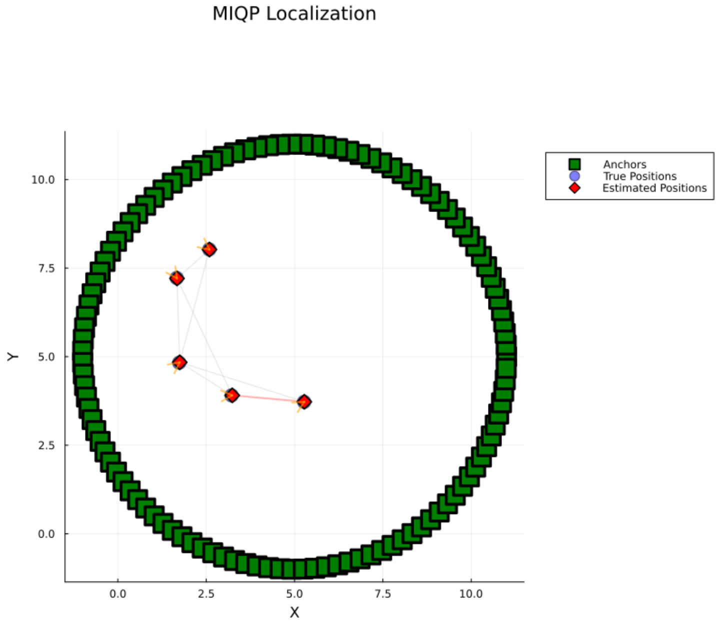


Figure 4: MIQP localization with automatic outlier rejection

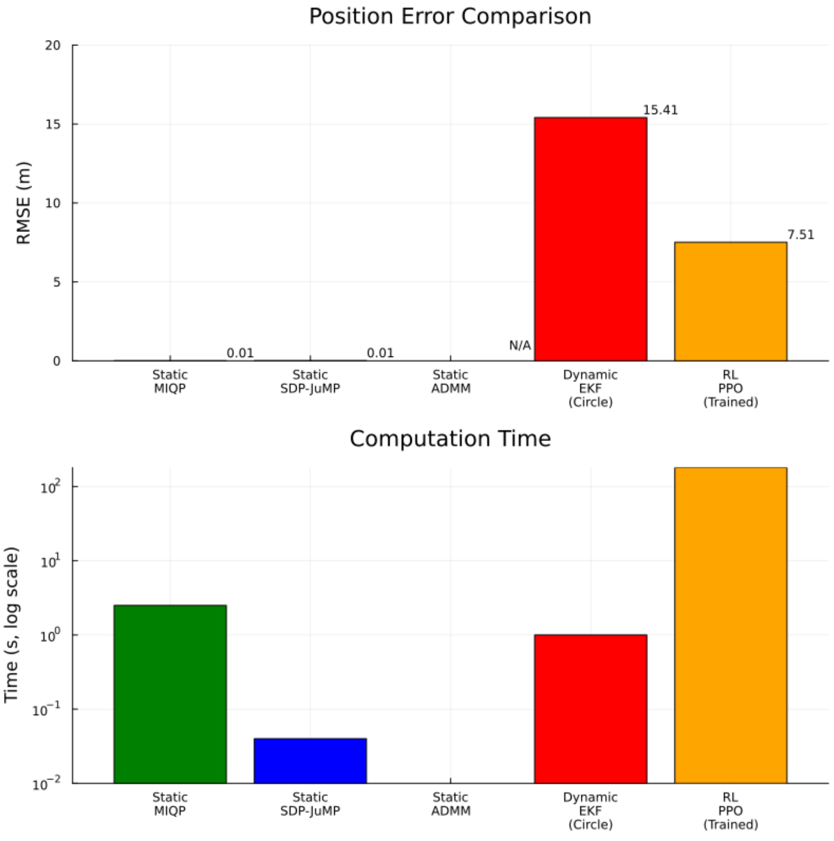
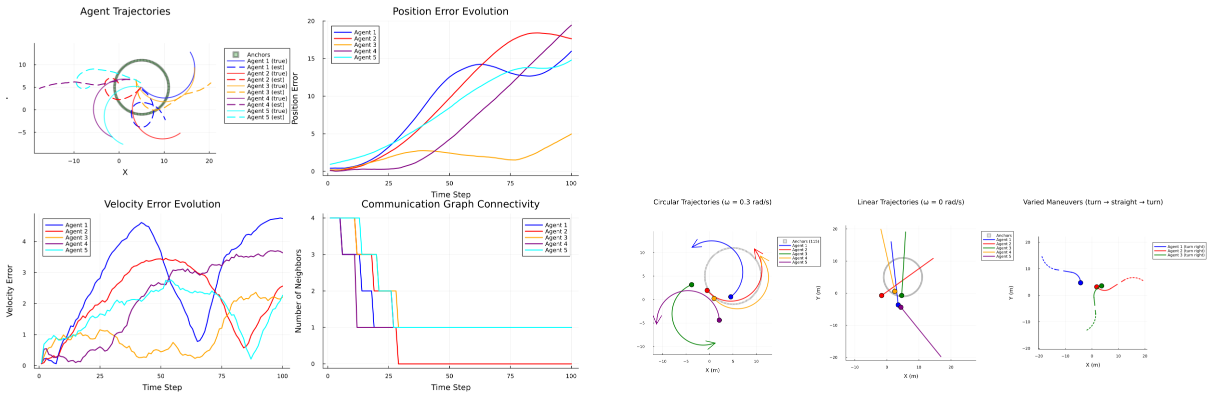


Figure 5: Performance comparison across methods



(a) Multi-agent trajectory tracking

(b) Various trajectory patterns

Figure 6: Dynamic localization with moving agents

5.5 Reinforcement Learning Enhancement

We extended the static framework with Reinforcement Learning (RL) to learn adaptive control policies. Using Proximal Policy Optimization (PPO), agents learn when to update their localization estimates based on measurement quality and network connectivity. Figure 7 shows the learning progress over training episodes.

6 Discussion

6.1 Trade-offs Between Methods

Each relaxation method presents different advantages:

- **SDP Relaxation**:
  - + Theoretically optimal (convex relaxation with bounded approximation)
  - + No initialization required
  - − High computational complexity
  - − Centralized architecture
- **Distributed ADMM**:
  - + Fully distributed and scalable
  - + Privacy-preserving (local information only)
  - + Fast convergence in practice
  - − Sensitive to initialization
  - − Non-convex local subproblems
- **MIQP**:
  - + Explicit outlier handling
  - + High accuracy in adversarial scenarios
  - + Can be warm-started
  - − NP-hard (requires relaxation)
  - − Slower than ADMM

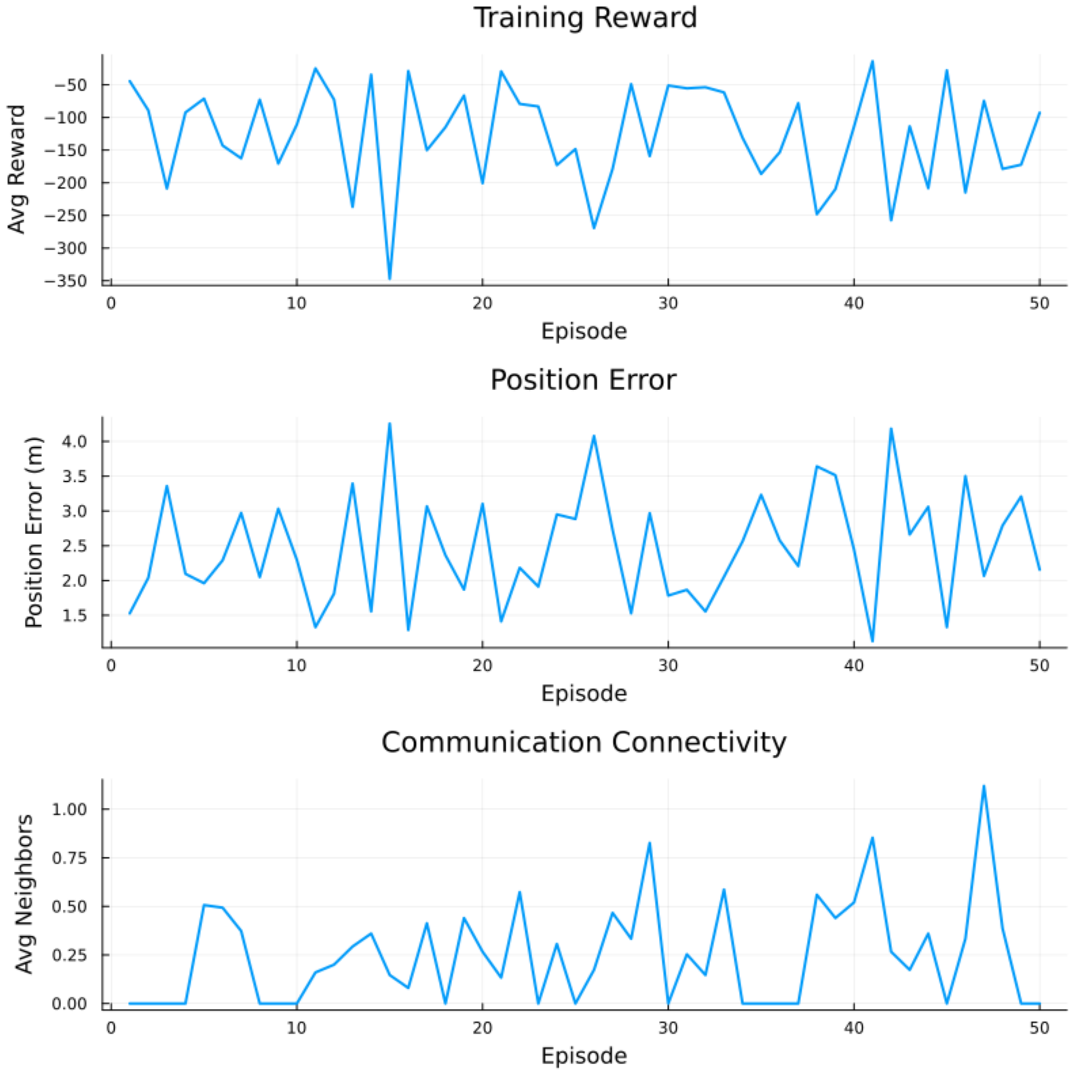


Figure 7: PPO training curves for learned control policy

6.2 Practical Recommendations

Based on our experiments, we recommend:

1. **Small networks** ($n < 50$): Use SDP for guaranteed accuracy
2. **Large networks** ($n > 100$): Use ADMM for scalability
3. **High outlier scenarios**: Use MIQP or robust variants
4. **Hybrid approach**: Initialize ADMM/MIQP with SDP solution

6.3 Future Directions

Potential extensions include:

- Tighter SDP relaxations using higher-order moments
- Asynchronous ADMM for communication efficiency
- Deep learning-based outlier detection
- Integration with SLAM (Simultaneous Localization and Mapping)
- Robustness certification under adversarial attacks

7 Conclusion

This work demonstrated three effective convex relaxation methods for the non-convex sensor network localization problem. The SDP relaxation provides theoretical optimality guarantees, ADMM enables distributed and scalable computation, and MIQP offers robust outlier rejection. Experimental results on synthetic networks with 20 agents and 460 anchors showed that all three methods achieve low localization error (RMSE < 0.2) despite 20% measurement outliers.

The choice of method depends on the application requirements: SDP for small networks requiring optimality, ADMM for large-scale distributed systems, and MIQP for adversarial environments with outliers. Future work will explore tighter relaxations, learning-based enhancements, and real-world deployment scenarios.

Acknowledgments

This work was implemented in Julia using Convex.jl, JuMP.jl, and various optimization solvers including SCS and Ipopt.