Appendix

Talha Bozkus and Urbashi Mitra

A. Mean vector and covariance matrix in state-estimation method

The mean vector $\boldsymbol{\mu}_{1,t}$ and covariance matrix $\boldsymbol{\Sigma}_{1,t}$ at time t are given as follows:

$$\boldsymbol{\mu}_{1,t} = \begin{pmatrix} \mu_{12,t} \\ \mu_{13,t} \end{pmatrix} \tag{1}$$

$$\boldsymbol{\Sigma}_{1,t} = \begin{pmatrix} \sigma^2_{12,t} & \sigma_{12,13,t} \\ \sigma_{13,12,t} & \sigma^2_{13,t} \end{pmatrix} \tag{2}$$

The elements of mean vector are computed as follows:

$$\mu_{12,t} = I_{ref} - 25 \cdot \log_{10}((x_{1,t} - x_{2,t})^2 + (y_{1,t} - y_{2,t})^2) \tag{3}$$

$$\mu_{13,t} = I_{ref} - 25 \cdot \log_{10}((x_{1,t} - x_{3,t})^2 + (y_{1,t} - y_{3,t})^2), \tag{4}$$

where $(x_{i,t}, y_{i,t})$ is the location of TX_i at time t, which are **known** as we condition on the knowledge of the states s_2 and s_3, and I_{ref} is the reference ARSS at 1m distance. The elements of $\boldsymbol{\Sigma}_{1,t}$ are computed as follows:

$$\sigma^2_{12,t} = \frac{1}{t}\sum_{k=1}^{t}(I_{1,k} - \mu_{12,t})^2 \tag{5}$$

$$\sigma_{12,13,t} = \sigma_{13,12,t} = \frac{1}{t}\sum_{k=1}^{t}(I_{1,k} - \mu_{12,t})(I_{1,k} - \mu_{13,t}) \tag{6}$$

$$\sigma^2_{13,t} = \frac{1}{t}\sum_{k=1}^{t}(I_{1,k} - \mu_{13,t})^2 \tag{7}$$

B. Search size for argmax operation

The argmax operator in (6) is applied to a subset of the state space determined by parameter Δ. Here, δ_0 denotes the initial distance between the true state and the first estimate. Given that each TX can move by Δ_L per step, the distance between TXs can increase by up to $2\Delta_L$ per step. Since the belief state resets every l steps, the distance can accumulate over l steps. To account for this, we use the modulo operation $t\%l$ to determine the distance since the last belief reset.

C. Total cost of information sharing

Assuming sharing each Q-function, cost, action, or state incurs one cost unit, we calculate the total sharing cost in four cases:

1) $U \rightarrow U$: TX_i sends $\frac{|\mathcal{S}_U|}{|\mathcal{S}|}|\mathcal{S}_i||\mathcal{A}_i|$ Q-functions to TX_1. With $N_T - 1$ TXs, the total cost is:

$$(N_T - 1)\frac{|\mathcal{S}_U|}{|\mathcal{S}|}|\mathcal{S}_i||\mathcal{A}_i|. \tag{8}$$

2) $U \rightarrow C$: TX_i shares its joint state estimate and belief probability with TX_1 (cost: $2(N_T - 1)$). TX_1 shares the joint action and Q-function with other TXs (cost: $2(N_T - 1)$). Total cost is:

$$4(N_T - 1). \tag{9}$$

3) $C \rightarrow U$: TX_i sends its joint state estimate and belief probability to TX_1 (cost: $2(N_T - 1)$). TX_1 shares the joint action with other TXs (cost: $N_T - 1$). TXs return their Q-functions and costs to TX_1 (cost: $2(N_T - 1)$). Notifying TX_1 of the coordinated state incurs a cost of 1, and TX_1 sharing this with $N_T - 1$ other TXs adds another $N_T - 1$. Total cost is:

$$6N_T - 5. \tag{10}$$

4) C → C: TX_i shares its joint state estimate and belief probability with TX_1 (cost: $2(N_T-1)$). TX_1 shares the joint action with other TXs (cost: N_T-1). TXs return their costs to TX_1 (cost: N_T-1). Notifying TX_1 of the coordinated state incurs a cost of 1, and TX_1 sharing this with N_T-1 other TXs adds another N_T-1. Total cost is:

$$5N_T-4. \tag{11}$$

We do not know the exact frequency of each case. Instead, we have $|\mathcal{S}_U|$ uncoordinated states and $|\mathcal{S}_C|$ coordinated states, with the algorithm running for T iterations. The time spent on uncoordinated states is $T\frac{|\mathcal{S}_U|}{|\mathcal{S}|}$, and on coordinated states is $T\frac{|\mathcal{S}_C|}{|\mathcal{S}|}$ on average. The average sharing cost in uncoordinated states is:

$$T\frac{|\mathcal{S}_U|}{|\mathcal{S}|}2(N_T-1)+(N_T-1)\frac{|\mathcal{S}_U|}{|\mathcal{S}|}|\mathcal{S}_i||\mathcal{A}_i|. \tag{12}$$

The average sharing cost in uncoordinated states is:

$$T\frac{|\mathcal{S}_C|}{|\mathcal{S}|}\frac{9(N_T-1)}{2} \tag{13}$$

Thus, the total sharing cost over T iterations is:

$$= T\frac{|\mathcal{S}_U|}{|\mathcal{S}|}2(N_T-1)+(N_T-1)\frac{|\mathcal{S}_U|}{|\mathcal{S}|}|\mathcal{S}_i||\mathcal{A}_i|+T\frac{|\mathcal{S}_C|}{|\mathcal{S}|}\frac{(11N_T-9)}{2} \tag{14}$$

$$\approx 2TN_T\frac{|\mathcal{S}_U|}{|\mathcal{S}|}+N_T\frac{|\mathcal{S}_U|}{|\mathcal{S}|}|\mathcal{S}_i||\mathcal{A}_i|+\frac{11TN_T}{2}\frac{|\mathcal{S}_C|}{|\mathcal{S}|} \tag{15}$$

$$= 2TN_T+N_T\frac{|\mathcal{S}_U|}{|\mathcal{S}|}|\mathcal{S}_i||\mathcal{A}_i|+\frac{7TN_T}{2}\frac{|\mathcal{S}_C|}{|\mathcal{S}|} \tag{16}$$

$$= N_T(2T+\frac{|\mathcal{S}_U|}{|\mathcal{S}|}|\mathcal{S}_i||\mathcal{A}_i|+\frac{7T}{2}\frac{|\mathcal{S}_C|}{|\mathcal{S}|}), \tag{17}$$

$$= O(N_T\max\{T,|\mathcal{S}_i||\mathcal{A}_i|\}), \tag{18}$$

where (15) follows by dropping the constants (since N is large), (16) follows as $|\mathcal{S}_C|+|\mathcal{S}_U|=|\mathcal{S}|$, (17) follows by organizing the terms, and (18) follows by the Big O notation and facts $|\mathcal{S}_C|\ll|\mathcal{S}|$ and $|\mathcal{S}_U|<|\mathcal{S}|$.

D. Parameter selection for Algorithm 1

- *Trajectory length* l: Selected by trial and error from $\{10, 20, \ldots, 50\}$.
- *Learning rate* α_t: Given by $\alpha_t=\frac{1}{1+\frac{t}{c_1}}$. The parameter c_1 is chosen based on state-space size as in [1].
- *Exploration probability* ϵ_t: Defined as $\epsilon_t=\max((c_2)^t,c_3)$. Parameters c_2 and c_3 are selected based on state-space size as in [1].
- *Update ratio* u_t: Given by $u_t=1-e^{-\frac{t}{c_4}}$. The parameter c_4 is chosen according to state-space size as in [1].
- *Grid Dimensions* L *and precision* Δ_L affect the state-space size. We select large L and small Δ_L to mirror real-world scenarios while remaining computationally feasible.
- *TX coverage areas:* Randomly chosen as long as at least one TX covers each grid. Hence, one can choose them randomly. By doing so, we make TXs with different quality.
- *ARSS Values and Conversion*

  We use $P_i=20$ dBm as the reference power level. The ARSS values for different transmitters depend on the total number of transmitters N_T as each transmitter receives contributions from the other N_T-1 transmitters. To determine

the ARSS values, we first compute the RSS values in the linear domain, multiply them by $N_T - 1$, and then convert the results back into the dBm domain. The formulas for the ARSS values are as follows:

$$I_{min} = 10 \cdot (N_T - 1) - 64 \text{ dBm} \quad \text{(at approximately 141 m maximum distance)} \tag{19}$$

$$I_{max} = 10 \cdot (N_T - 1) - 10 \text{ dBm} \quad \text{(at approximately 1 m maximum distance)} \tag{20}$$

$$I_{thr} = 10 \cdot (N_T - 1) - 45 \text{ dBm} \quad \text{(at approximately 25 m maximum distance)} \tag{21}$$

$$I_{ref} = 10 \cdot (N_T - 1) - 10 \text{ dBm} \quad \text{(at approximately 1 m maximum distance)} \tag{22}$$

Here, the precision value for quantization is fixed at $\Delta_I = 10$.

- *Simulated Local ARSS Measurements*

To simulate the local ARSS measurements, we use a log-distance path loss model, incorporating noise in the distance measurements. The formula for the simulated measurements $I(d)$ is given by:

$$I(d) = I_{ref} - \sum_{i=1}^{N_T-1} 25 \log(d_i + n_i), \quad n_i \sim \text{unif}[-2\Delta_L, 2\Delta_L], \tag{23}$$

where: d_i is the distance between TX_1 and TX_i and n_i is the noise, uniformly distributed in the interval $[-2\Delta_L, 2\Delta_L]$. This formula extends the RSS to distance conversion by summing up the contributions from $N_T - 1$ TXs to account for the ARSS scenario. The term $25 \log(d_i + n_i)$ reflects the path loss model, with the added noise term n_i simulating measurement inaccuracies.

- *Belief Vector Initialization*

The belief vector $\mathbf{b}_{1,0}(s_2, s_3)$ represents our belief about the system's state at time $t = 0$. It is divided into two components:

1) Belief Over Joint Locations: This is denoted as $b_{1,0}((x_2, y_2), (x_3, y_3))$, which expresses our belief about the locations of TX_2 and TX_3 relative to TX_1.
2) Belief Over Joint ARSS Measurements: This is denoted as $b_{1,0}(I_2, I_3)$, which expresses our belief about the ARSS measurements from TX_2 and TX_3.

Assuming the initial location of TX_1 is (x_0, y_0), the distances from TX_1 to TX_2 and TX_3 are given by $d_2 = \|(x_2, y_2) - (x_0, y_0)\|_2$ and $d_3 = \|(x_3, y_3) - (x_0, y_0)\|_2$, respectively.

1) Belief Over Locations: We approximate the belief over the joint locations (x_2, y_2) and (x_3, y_3) as:

$$b_{1,0}((x_2, y_2), (x_3, y_3)) \approx e^{-d_2 - d_3} \tag{24}$$

This exponential term reflects a decreasing belief with increasing distances.

2) Belief Over ARSS Measurements: The belief over the measurements I_2 and I_3 is modeled as:

$$b_{1,0}(I_2, I_3) \approx N\left(I_2 \mid I_{1,0} \frac{d_2^{-1}}{d_2^{-1} + d_3^{-1}}, 1\right) \cdot N\left(I_3 \mid I_{1,0} \frac{d_3^{-1}}{d_2^{-1} + d_3^{-1}}, 1\right). \tag{25}$$

where $N(\cdot \mid \mu, \sigma^2)$ denotes the normal distribution with mean μ and variance σ^2. Here, $I_{1,0}$ is the initial measurement, and the weights $\frac{d_2^{-1}}{d_2^{-1}+d_3^{-1}}$ and $\frac{d_3^{-1}}{d_2^{-1}+d_3^{-1}}$ represent the relative influence of each distance. Herein, we assume that the ARSS measurements are normally distributed around a mean that is weighted by the inverse of the distances. This reflects the intuition that closer TXs have a more contribution on ARSS.

3) Combined Belief Vector: The overall belief vector is:

$$\mathbf{b}_{1,0}(s_2, s_3) = \frac{b_{1,0}((x_2, y_2), (x_3, y_3)) \cdot b_{1,0}(I_2, I_3)}{\sum_{(x', y', I')} b_{1,0}((x_2', y_2'), (x_3', y_3')) \cdot b_{1,0}(I_2', I_3')}. \tag{26}$$

This combines both components, normalized by the sum of all possible combinations of locations and measurements.

E. Details for other algorithms

- In IQ, each TX runs independent Q-learning, ignoring other agents. The joint Q-table is formed by concatenating individual Q-tables, from which the joint policy is derived. The Q-learning parameters, including $\alpha_t, \epsilon_t, \gamma$, follow the same structure as in our algorithm and are optimized by trial and error.
- In HQ, the following update rule is used:

$$\begin{aligned} \delta &\leftarrow c_i(s_i, a_i) + \gamma \min_{a_i' \in \mathcal{A}} Q_i(s_i', a_i') - Q_i(s_i, a_i) \\ Q_i\left(s_i, a_i\right) &\leftarrow \begin{cases} Q_i\left(s_i, a_i\right) + \alpha\delta & \text{if } \delta \leq 0 \\ Q_i\left(s_i, a_i\right) + \beta\delta & \text{else} \end{cases} \end{aligned} \tag{27}$$

  with α and β are chosen time-varying as $\alpha_t = \frac{1}{1+\frac{t}{c_1}}$ and $\beta_T = \frac{1}{1+\frac{t}{c_2}}$, and $c_1 > c_2$ such that the learning rate is higher for cost-reducing experiences. The other parameters and the joint Q-table construction are done as in IQ.
- In PSAQ, the asking and giving probabilities are set as in [2] with $v_a = 0.2$ and $v_b = 1$. The budgets are $b^i_{ask} = b^i_{give} \sim \mathcal{N}(3000, 50)$. The other parameters and the joint Q-table construction are done as in IQ.
- In SCQ, the Q-learning parameters are chosen as in IQ. We use centralized learning in coordinated states with full access to joint states, actions, and costs.
- In MEMQ, we treat the overall multi-agent network as a single agent. The parameters follow the same structure as in [1] and are optimized by trial-error.

REFERENCES

[1] T. Bozkus and U. Mitra, "Multi-timescale ensemble *q*-learning for markov decision process policy optimization," *IEEE Transactions on Signal Processing*, vol. 72, pp. 1427–1442, 2024.

[2] C. Zhu, H.-F. Leung, S. Hu, and Y. Cai, "A q-values sharing framework for multi-agent reinforcement learning under budget constraint," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 15, no. 2, pp. 1–28, 2021.