ARO HW4 2024

GraphSLAM

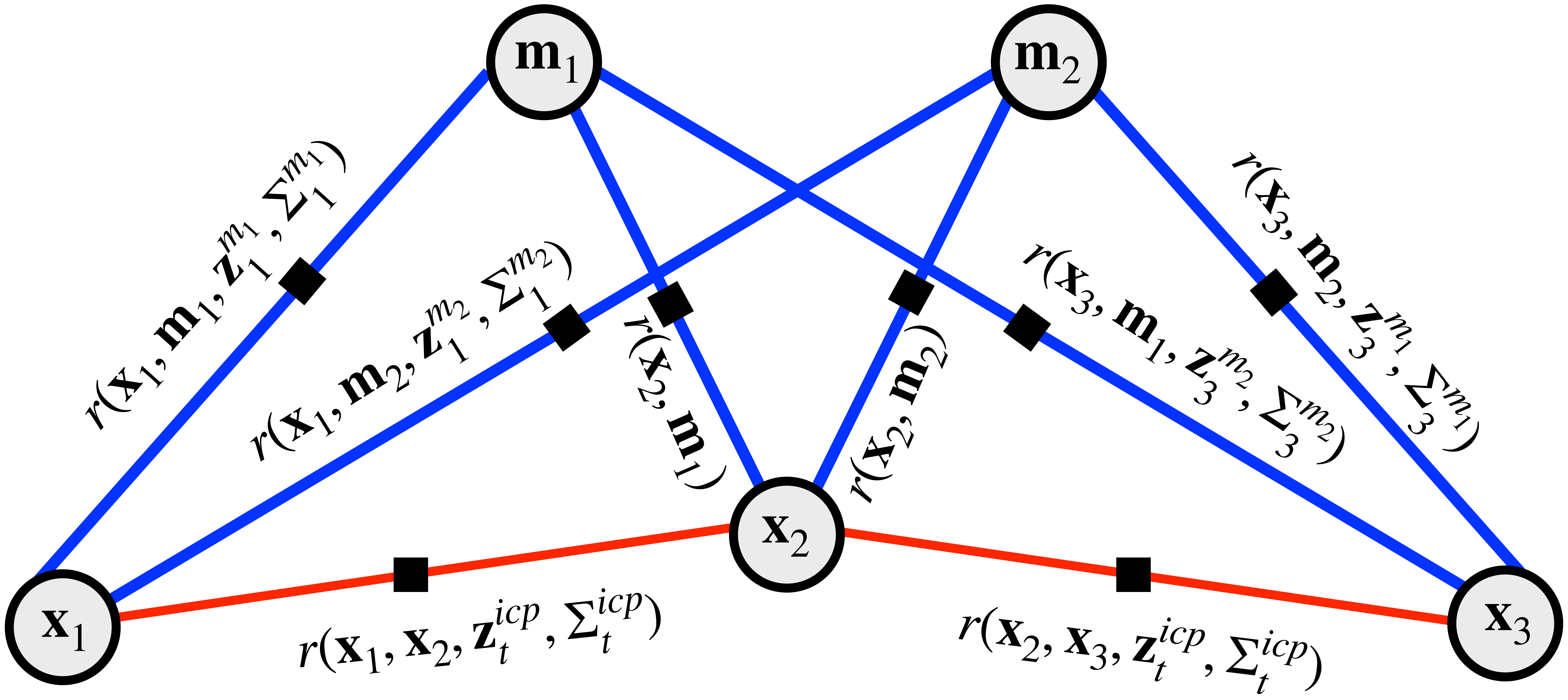
HW4 factorgraph

ICP odometry — $r(\mathbf{x}_t, \mathbf{x}_{t+1})$

3D marker(s) — $r(\mathbf{m}_j, \mathbf{x}_t)$

$$\mathbf{x}_1^\star, \dots \mathbf{x}_T^\star, \mathbf{m}_1^\star, \dots \mathbf{m}_J^\star = \arg\min_{\substack{\mathbf{x}_1\dots\mathbf{x}_T \\ \mathbf{m}_1\dots\mathbf{m}_J}} \sum_t \|\mathrm{w2r}(\mathbf{x}_{t+1}, \mathbf{x}_t) - \mathbf{z}_t^{icp}\|^2_{\Sigma_t^{icp}} + \sum_{t,j} \|\mathrm{w2r}(\mathbf{m}_j, \mathbf{x}_t) - \mathbf{z}_t^{m_j}\|^2_{\Sigma_t^{m_j}}$$

$$= \arg\min_{\mathbf{x}_1\dots\mathbf{x}_T} \sum_t \|r(\mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{z}_t^{icp}, \Sigma_t^{icp})\|^2 + \sum_{t,j} \|r(\mathbf{x}_t, \mathbf{m}_j, \mathbf{z}_t^{m_j}, \Sigma_t^{m_j})\|^2$$



$$r(\mathbf{x}_t, \mathbf{m}_j, \mathbf{z}_t^{m_j}, \Sigma_t) = \mathrm{w2r}(\mathbf{m}_j, \mathbf{x}_t) - \mathbf{z}_t = \Sigma_t^{-\frac{1}{2}} \cdot \begin{bmatrix} +\cos\theta_t \cdot (m_j^x - x_t) + \sin\theta_t \cdot (m_j^y - y_t) - z_t^x \\ -\sin\theta_t \cdot (m_j^x - x_t) + \cos\theta_t \cdot (m_j^y - y_t) - z_t^y \\ m_j^\theta - \theta_t - z_t^\theta \end{bmatrix}$$

$$r(\mathbf{x}_t, \mathbf{m}_j, \mathbf{z}_t^{m_j}, \Sigma_t) : \mathbb{R}^6 \to \mathbb{R}^3 \qquad \mathbf{J}_{3\times 6} = \left[\frac{\partial r(\mathbf{x}_t, \mathbf{m}_j)}{\partial \mathbf{x}_t}, \frac{\partial r(\mathbf{x}_t, \mathbf{m}_j)}{\partial \mathbf{m}_j}\right]$$

$$\mathbf{J}_{3\times 6} = \Sigma_t^{-\frac{1}{2}} \cdot \begin{array}{cccccc} \frac{\partial}{\partial x_t} & \frac{\partial}{\partial y_t} & \frac{\partial}{\partial \theta_t} & \frac{\partial}{\partial m_t^x} & \frac{\partial}{\partial m_t^y} & \frac{\partial}{\partial m_t^\theta} \end{array}$$

$$\begin{bmatrix} -\cos\theta_t & -\sin\theta_t & -\sin\theta_t \cdot (x_{t+1} - x_t) + \cos\theta_t \cdot (y_{t+1} - y_t) & \cos\theta_t & \sin\theta_t & 0 \\ +\sin\theta_t & -\cos\theta_t & -\cos\theta_t \cdot (x_{t+1} - x_t) - \sin\theta_t \cdot (y_{t+1} - y_t) & \text{-}\sin\theta_t & \cos\theta_t & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix}$$

$$r(\mathbf{x}_t, \mathbf{m}_j, \mathbf{z}_t^{m_j}, \Sigma_t) = \mathrm{w2r}(\mathbf{m}_j, \mathbf{x}_t) - \mathbf{z}_t = \Sigma_t^{-\frac{1}{2}} \cdot \begin{bmatrix} +\cos\theta_t \cdot (m_j^x - x_t) + \sin\theta_t \cdot (m_j^y - y_t) - z_t^x \\ -\sin\theta_t \cdot (m_j^x - x_t) + \cos\theta_t \cdot (m_j^y - y_t) - z_t^y \\ m_j^\theta - \theta_t - z_t^\theta \end{bmatrix}$$

$$r(\mathbf{x}_t, \mathbf{m}_j, \mathbf{z}_t^{m_j}, \Sigma_t) : \mathbb{R}^6 \rightarrow \mathbb{R}^3$$

$$\mathbf{J}_{3\times 6} = \left[\frac{\partial r(\mathbf{x}_t, \mathbf{m}_j)}{\partial \mathbf{x}_t}, \frac{\partial r(\mathbf{x}_t, \mathbf{m}_j)}{\partial \mathbf{m}_j}\right]$$

```
def res(x, m, z, s_inv):
    # convert vector m into wcf and compute its residual wrt z
    # result is 3x1 vector
   return res

def res_jac(x, m, z, s_inv):
    # implement 3x6 jacobian of the residual res(m, x, z, s_inv) wrt (m,x)
    return J
```