

# Formulation of the problem

Chang Liu

## I. INTRODUCTION

My work can different from Charrow's work in the following aspects: 1. Instead of fitting each particle with a GMM, I generate a GMM with much smaller component number for planning purpose. 2. In the planning, I incorporate the effects of limited FOV. 3. In the path planning, I use MPC that may incorporate uncertainties.

I should mention that, strategies that only updates GMM weights are not suitable, since the GMM here is not for mode that is fixed, but for dynamically changing states. Therefore, it is important to update the mean and covariance.

## II. PROBLEM FORMULATION

### A. Robot and Target Motion Model

Unicycle motion model for the mobile robot:

$$z_{k+1} = h(z_k, u_k^r), \quad (1)$$

where

$$f(z_k, u_k^r) = z_k + \begin{bmatrix} \cos \theta_k^r \Delta t & 0 \\ \sin \theta_k^r \Delta t & 0 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} u_k^r.$$

Motion model of the target:

$$x_{k+1} = f(x_k) + w_k, \quad w_t \sim \mathcal{N}(0, Q) \quad (2)$$

$$(3)$$

### B. Modeling Sensing Domain

Sensor sensing domain is represented as  $\mathcal{F}_k = \{[x_{1,k}, x_{2,k}] \in \mathbb{R}^2 \mid \|v\|_2 \leq r, \angle v \in [\theta_1, \theta_2]\}$ , where  $v = [x_{1,k} - z_{1,k}, x_{2,k} - z_{2,k}]$ .

### C. Sensor Measurement Model

Measurement model:

$$y_k = g(x_k) + v_k, \quad v_k \sim \begin{cases} \mathcal{N}(0, R) & \text{if } \gamma_k = 1 \\ \mathcal{N}(0, \sigma^2 I) & \text{if } \gamma_k = 0 \end{cases}, \quad (4)$$

$$\gamma_k = \mathbb{1}_{\{x_k \in \mathcal{F}_k\}} \quad (5)$$

## III. MPC-BASED PATH PLANNING

### A. EKF with Limited Sensing Domain

$$\hat{x}_{k+1|k}^t = f(\hat{x}_{k|k}^t) \quad (6a)$$

$$P_{k+1|k} = A_k^i P_{k|k}^i A_{k+1}^{i'} + Q \quad (6b)$$

$$K_{k+1}^i = P_{k+1|k}^i C_{k+1}^{i'} (C_{k+1}^{i'} P_{k+1|k}^{i'} C_{k+1}^{i'} + R)^{-1} \quad (6c)$$

$$\hat{x}_{k+1|k+1}^i = \hat{x}_{k+1|k}^i + \gamma_{k+1} K_{k+1}^i (y_{k+1} - h(\hat{x}_{k+1|k}^t)) \quad (6d)$$

$$P_{k+1|k+1}^i = P_{k+1|k}^i - \gamma_{k+1} K_{k+1}^i C_{k+1}^{i'} P_{k+1|k}^i, \quad (6e)$$

where  $A_k^i = \frac{\partial f}{\partial x}|_{x=\hat{x}_{k|k}^i}$  and  $C_{k+1}^i = \frac{\partial g}{\partial x}|_{x=\hat{x}_{k+1|k}^i}$ . The  $\hat{x}_{k|k}^t$  and  $P_{k|k}$  represent the estimated target position and covariance matrix. For notational simplicity, we define  $b_k = [\hat{x}_{k|k}^t, P_{k|k}]$  and let  $b_{k+1} = g(b_k, u_k^r)$  represent the Kalman filter defined in Eq. (6).

$\gamma$  is approximated by

$$\gamma_k \approx \frac{1}{1 + \alpha_1 \|[x_{1,k}, x_{2,k}] - [z_{1,k}, z_{2,k}]\|_2^2} \times \frac{1}{1 + \exp \left\{ -\alpha_2 (\cos(\theta_k^r - \tilde{\theta}_k) - \cos(\theta_0)) \right\}}, \quad (7)$$

where  $\tilde{\theta}_k = \angle([x_{1,k}, x_{2,k}] - [z_{1,k}, z_{2,k}])$  is the direction angle from the sensor position to target position;  $\theta_0 = \frac{\theta_2 - \theta_1}{2}$  is half of the sensing angle;  $\alpha_1$  and  $\alpha_2$  are tuning parameters that controls the shape of the function. Eq. (7) can be interpreted as follows: when the robot is close to the target, it is more likely that the target can be detected; besides, the closer the target direction aligns with the center direction of the sensor, the higher possibility that the target will get detected.

### B. Path Planning for Target Search and Tracking

The MPC-based path planner with planning horizon  $N$  can be formulated as:

$$\min_{u_{1:N}} J(b_{1:N+1}, u_{1:N}) \quad (8a)$$

$$\text{s.t. } z_{k+1} = f(z_k, u_k^r), \quad (8b)$$

$$b_{k+1} = g(b_k, u_k^r), \quad (8c)$$

$$z_{k+1} \in \mathcal{X}, \quad u_{k+1}^r \in \mathcal{U}, \quad (8d)$$

$$k = 1, \dots, N, \quad (8e)$$

The objective function is

$$J(b_{1:N+1}) = \sum_{k=1}^{N+1} H(b_k) \quad (9)$$

$$\approx - \sum_{k=1}^{N+1} \sum_{i=1}^L w_i \log b_k^i. \quad (10)$$

The approximation is the 0-order approximation of entropy.

### C. Possible linearization in the iterative planning process

- 1) target motion matrix A
- 2) sensor measurement matrix C
- 3) initial solution
- 4) approximate the sensor boundary
- 5) linearize robot motion model

#### *D. estimation-planning framework*

In the estimation based on the actual measurement, use the particle filter to incorporate nonlinearity and use E-M at each step to fit a GMM. mention that we can use other advanced technologies to update GMM, but here we just use a simple one.

In the planning, use  $\gamma$  to update the estimation. This makes sense since if we use pseudo-measurement to update GMM, it only reinforces/bootstrapping the current GMM, bringing no new info. So it makes sense to just consider the entropy of the uncertainty.

In the MPC, I may need to include the uncertainty into the obj.

#### IV. POSSIBLE EXTENSIONS

- 1) GSF with good weight and component number update law
  - a) GMM-PF: update PF and GMM at the same time, not running E-M to fit GMM for PF. One benefit of this is that we can always check the difference between PF and GMM and generate a better GMM if difference is big.
  - b) when updating GMM, can we treat  $\gamma = 0$  as a rare event? Will Dirichlet process help?
  - c) treat  $\gamma = 0$  and  $\gamma = 1$  as hybrid system. Then use GMM to model the mode switch? I think a little bit about this but seems difficult to do.
- 2) efficient computation of the objective function
- 3) how to represent and compute  $\gamma$  (incorporating  $b_t$  or just using a point estimate (e.g. MAP))
- 4) incorporate negative info in a better way than  $\gamma$ .
- 5) the way to do iterative planning: updating  $\gamma$  in SQP or outside SQP; whether updating  $w$ . if not updating, can I obtain an upper bound of the error?
- 6) control of nonholonomic vehicle
- 7) make the problem a cvx optimization or some other form (e.g., proximal gradient) to better utilize the form of the problem.

1,3,4 are the possible main contributions of the work.