"Safe Corridor-Based MPC for Follow-Ahead and Obstacle Avoidance of Mobile Robot in Cluttered Environments" Detailed Algorithms

A. Setup of Convex Polygonal Safe Region

Algorithm 1: SafeRegionSetup

Input: $\mathcal{C}$, $\mathrm{d_{cor}}$, ${}^{\mathrm{G}}P_{\mathrm{obs}}$
Output: $\mathcal{C}$
`// Initialize the Convex Safe Region`
Initialize all $\mathcal{C}.d_i$ as $\mathrm{d_{cor}}$;
for $\mathbf{p}$ in ${}^{\mathrm{G}}P_{\mathrm{obs}}$ **do**
    Get the index i of the sector which $\mathbf{p}$ belongs to;
    $\mathcal{C}.d_i = \min(\mathcal{C}.d_i,\ \mathbf{e}_i^T(\mathbf{p} - \mathcal{C}.\mathbf{p}_{\mathrm{pc}}) - \mathrm{r_r})$;
$\mathcal{C}.\mathbf{A}_{\mathrm{cor}} = \left[\mathbf{e}_1, \cdots, \mathbf{e}_{\mathrm{n_{bin}}}\right]^T$;
$\mathcal{C}.\mathbf{b}_{\mathrm{cor}} = \mathcal{C}.\mathbf{d} + \mathcal{C}.\mathbf{A}_{\mathrm{cor}}\mathcal{C}.\mathbf{p}_{\mathrm{pc}}$;
`// Extend the Convex Safe Region`
Initialize a Boolean matrix $\mathbf{O} = \{o_{i,j}\}_{\mathrm{n_{bin}} \times |{}^{\mathrm{G}}P_{\mathrm{obs}}|}$;
for j, $\mathbf{p}$ in enumerate$({}^{\mathrm{G}}P_{\mathrm{obs}})$ **do**
    for i in range$(0, \mathrm{n_{bin}})$ **do**
        $o_{i,j} = \mathbf{e}_i^T(\mathbf{p} - \mathcal{C}.\mathbf{p}_{\mathrm{pc}}) \geq 0$;
for i in range$(0, \mathrm{n_{bin}})$ **do**
    $d_{\mathrm{tmp},i} = \mathrm{d_{cor}}$;
    for j, $\mathbf{p}$ in enumerate$({}^{\mathrm{G}}P_{\mathrm{obs}})$ **do**
        if $o_{i,j}$ **and** sum$(o_{\cdot,j}) == 1$ **then**
            $d_{\mathrm{tmp},i} = \min(d_{\mathrm{tmp},i},\ \mathbf{e}_i^T(\mathbf{p} - \mathcal{C}.\mathbf{p}_{\mathrm{pc}}) - \mathrm{r_r})$;
    $\mathcal{C}.\mathbf{b}_{\mathrm{cor}} = \mathbf{d}_{\mathrm{tmp}} + \mathcal{C}.\mathbf{A}_{\mathrm{cor}}\mathcal{C}.\mathbf{p}_{\mathrm{pc}}$;
    for j, $\mathbf{p}$ in enumerate$({}^{\mathrm{G}}P_{\mathrm{obs}})$ **do**
        $o_{i,j} = \mathbf{e}_i^T(\mathbf{p} - \mathcal{C}.\mathbf{p}_{\mathrm{pc}}) \geq 0$;

B. Iterative Human Route Search Algorithm

Algorithm 2: GetSubnode

Input: k, $\mathcal{C}$, ν, $\mathrm{d_{cor}}$, $\mathrm{n_1}$, $\mathrm{n_2}$, $\mathcal{L}$, Θ, ${}^{\mathrm{G}}P_{\mathrm{obs}}$
Output: $\mathcal{C}$
$s_{k+1,0} = \varnothing$, $\theta_{k+1,0} = \varnothing$, $s_{k+1,1} = \varnothing$, $\theta_{k+1,1} = \varnothing$;
$a = \lfloor \mathrm{n_{bin}}\theta_{\mathrm{des},k+1}/2\pi \rfloor$;
$b = \lfloor \mathrm{n_{bin}}\mathcal{C}.\theta/2\pi \rfloor$;
`// Safe Region Setup`
$\mathcal{C}$ = SafeRegionSetup$(\mathcal{C}, \mathrm{d_{cor}}, {}^{\mathrm{G}}P_{\mathrm{obs}})$;
`// Iterative Human Route Search`
if $k <= |\mathcal{L}|$ **then**
    if $\mathcal{C}.d_a \geq \nu\mathrm{d_{cor}}$ **then**
        `// 1.Follow the desired direction`
        $s_{k+1,0}$ = FORWARD;
        if $\mathcal{C}.s ==$ FORWARD **then**
            $\theta_{k+1,0} = \theta_{\mathrm{des},k+1}$;
        else
            $\theta_{k+1,0} = \xi\theta_{\mathrm{des},k+1} + (1-\xi)\mathcal{C}.\theta$;
    else
        if $\mathcal{C}.d_b \geq \nu\mathrm{d_{cor}}$ **then**
            `// 2.Follow the previous direction`
            $s_{k+1,0} = \mathcal{C}.s$, $\theta_{k+1,0} = \mathcal{C}.\theta$;
        else
            `// 3.Take a smaller detour`
            $s_{k+1,0}$, $\theta_{k+1,0}$, $s_{k+1,1}$, $\theta_{k+1,1}$ = SearchBothSides$(\mathcal{C}, \nu, \mathrm{d_{cor}}, 1, \mathrm{n_1})$;
            if $s_{k+1,0} \neq \varnothing$ **and** $s_{k+1,1} \neq \varnothing$ **then**
                $\Delta\theta_{\mathrm{left}} = \theta_{k+1,0} - \mathcal{C}_k.\theta_k$;
                $\Delta\theta_{\mathrm{right}} = \theta_{k+1,1} - \mathcal{C}_k.\theta_k$;
                if $|\Delta\theta_{\mathrm{left}}| < |\Delta\theta_{\mathrm{right}}|$ **then**
                    $s_{k+1,1} = \varnothing$, $\theta_{k+1,1} = \varnothing$;
                else
                    $s_{k+1,0} = \varnothing$, $\theta_{k+1,0} = \varnothing$;
            if $s_{k+1,0} == \varnothing$ **and** $s_{k+1,1} == \varnothing$ **then**
                `// 4.Take a larger detour`
                $s_{k+1,0}$, $\theta_{k+1,0}$, $s_{k+1,1}$, $\theta_{k+1,1}$ = SearchBothSides$(\mathcal{C}, \nu, \mathrm{d_{cor}}, \mathrm{n_1}, \mathrm{n_2})$;
            if $s_{k+1,0} == \varnothing$ **and** $s_{k+1,1} == \varnothing$ **then**
                `// 5.Stop walking`
                $l_{\mathrm{des},k} = 0$, $s_{k+1,0}$ = STOP, $\theta_{k+1,0} = \mathcal{C}.\theta$;
    for i in range$(0, 2)$ **do**
        if $s_{k+1,i} \neq \varnothing$ **then**
            $\mathcal{C}.sub_i.\mathbf{p}_{\mathrm{pc}} = \mathcal{C}.\mathbf{p}_{\mathrm{pc}} + l_{\mathrm{des},k}\left[\cos(\theta_{k+1,i}), \sin(\theta_{k+1,i})\right]^T$;
            $\mathcal{C}.sub_i.\theta = \theta_{k+1,i}$, $\mathcal{C}.sub_i.s = s_{k+1,i}$;
            $\mathcal{C}$ = GetSubnode$(k+1, \mathcal{C}.sub_i, \nu, \mathrm{d_{cor}}, \mathrm{n_1}, \mathrm{n_2}, \mathcal{L}, \Theta, {}^{\mathrm{G}}P_{\mathrm{obs}})$;

Algorithm 3: SearchBothSides

Input: $\mathcal{C}$, ν, $\mathrm{d_{cor}}$, n_{start}, n_{end}
Output: $s_{k+1,0}$, $\theta_{k+1,0}$, $s_{k+1,1}$, $\theta_{k+1,1}$
$s_{k+1,0} = \varnothing$, $\theta_{k+1,0} = \varnothing$, $s_{k+1,1} = \varnothing$, $\theta_{k+1,1} = \varnothing$;
if $\mathcal{C}.s \neq$ RIGHT **then**
    for i in range$(n_{\mathrm{start}}, n_{\mathrm{end}})$ **do**
        $i_{\mathrm{l}} = \mathrm{mod}(a - i + \mathrm{n_{bin}},\ \mathrm{n_{bin}})$;
        if $\mathcal{C}.d_{i_{\mathrm{l}}} \geq \nu\mathrm{d_{cor}}$ **then**
            $s_{k+1,0}$ = LEFT;
            $\theta_{k+1,0} = \xi\pi(2i_{\mathrm{l}} - 1)/\mathrm{n_{bin}} + (1-\xi)\mathcal{C}.\theta$;
if $\mathcal{C}.s \neq$ LEFT **then**
    for i in range$(-n_{\mathrm{start}}, -n_{\mathrm{end}}, -1)$ **do**
        $i_{\mathrm{r}} = \mathrm{mod}(a + i,\ \mathrm{n_{bin}})$;
        if $\mathcal{C}.d_{i_{\mathrm{r}}} \geq \nu\mathrm{d_{cor}}$ **then**
            $s_{k+1,1}$ = RIGHT;
            $\theta_{k+1,1} = \xi\pi(2i_{\mathrm{r}} - 1)/\mathrm{n_{bin}} + (1-\xi)\mathcal{C}.\theta$;

C. Obtain the Most Similar Branch of Route Binary Tree

Algorithm 4: GetBestBranch

Input: k, $\mathcal{C}$, J_{tvl}, N, $\mathcal{P}_{\mathrm{r}} = \left\{{}^{\mathrm{G}}\mathbf{p}_{\mathrm{r},k_0-1}(k_0+k-1) | k = 0, 1, ..., \mathrm{N}\right\}$
Output: $\mathcal{C}$, J_{tvl}
if $k <= \mathrm{N}$ **and** $\mathcal{C} \neq \varnothing$ **then**
    $J_{\mathrm{tvl}} = J_{\mathrm{tvl}} + \max\left[\mathbf{A}_{\mathrm{cor}}(k)\,{}^{\mathrm{G}}\mathbf{p}_{\mathrm{r},k_0-1}(k-1) - \mathbf{b}_{\mathrm{cor}}(k)\right]$;
    $\mathcal{C}_{\mathrm{best}} = \varnothing$;
    $J_{\mathrm{best}} = 0$;
    for i in range$(0, 2)$ **do**
        if $\mathcal{C}.sub_i \neq \varnothing$ **then**
            $\mathcal{C}_{\mathrm{tmp}}$, J_{tmp} = GetBestBranch$(k+1, \mathcal{C}.sub_i, J_{\mathrm{tvl}}, \mathrm{N}, \mathcal{P}_{\mathrm{r}})$;
            if $\mathcal{C}_{\mathrm{best}} == \varnothing$ **or** $J_{\mathrm{tmp}} < J_{\mathrm{best}}$ **then**
                $J_{\mathrm{best}} = J_{\mathrm{tmp}}$;
                $\mathcal{C}_{\mathrm{best}} = \mathcal{C}_{\mathrm{tmp}}$;
    if $\mathcal{C}_{\mathrm{best}} \neq \varnothing$ **then**
        $J_{\mathrm{tvl}} = J_{\mathrm{best}}$;
        $\mathcal{C}.sub_0 = \mathcal{C}_{\mathrm{best}}$;
        $\mathcal{C}.sub_1 = \varnothing$;

In Algorithm 4, the similarity cost $J_{\mathrm{tvl}} = \sum_{k=k_0}^{k_0+\mathrm{N}} \max\left\{\mathbf{A}_{\mathrm{cor}}(k)\,{}^{\mathrm{G}}\mathbf{p}_{\mathrm{r},k_0-1}(k-1) - \mathbf{b}_{\mathrm{cor}}(k)\right\}$ not only reflects the distance between the human and the robot, but also the visibility of the robot seeing from the position of the followed human. For the first point, when the robot planned position ${}^{\mathrm{G}}\mathbf{p}_{\mathrm{r},k_0-1}(k-1)$ get close to the human, i.e. the center reference point $\mathbf{p}_{\mathrm{pc}}(k)$, straightly, the the corresponding term

$$\begin{aligned} &\max\left\{\mathbf{A}_{\mathrm{cor}}(k)\,{}^{\mathrm{G}}\mathbf{p}_{\mathrm{r},k_0-1}(k-1)-\mathbf{b}_{\mathrm{cor}}(k)\right\}\\ =&\max\left\{\mathbf{A}_{\mathrm{cor}}(k)\left({}^{\mathrm{G}}\mathbf{p}_{\mathrm{r},k_0-1}(k-1)-\mathbf{p}_{\mathrm{pc}}(k)\right)\right\}\\ =&\max\left\{\mathbf{A}_{\mathrm{cor}}(k)\left\|{}^{\mathrm{G}}\mathbf{p}_{\mathrm{r},k_0-1}(k-1)-\mathbf{p}_{\mathrm{pc}}(k)\right\|\,{}^{\mathrm{G}}\mathbf{e}_{\mathrm{cor}}(k)\right\}\\ =&\left\|{}^{\mathrm{G}}\mathbf{p}_{\mathrm{r},k_0-1}(k-1)-\mathbf{p}_{\mathrm{pc}}(k)\right\|\max\left\{\mathbf{A}_{\mathrm{cor}}(k)\,{}^{\mathrm{G}}\mathbf{e}_{\mathrm{cor}}(k)\right\} \end{aligned} \tag{1}$$

will become smaller, where ${}^{\mathrm{G}}\mathbf{e}_{\mathrm{cor}}(k)=\left({}^{\mathrm{G}}\mathbf{p}_{\mathrm{r},k_0-1}(k-1)-\mathbf{p}_{\mathrm{pc}}(k)\right)/\left\|{}^{\mathrm{G}}\mathbf{p}_{\mathrm{r},k_0-1}(k-1)-\mathbf{p}_{\mathrm{pc}}(k)\right\|$. For the second point, when the robot planned position is in the convex safe region, the corresponding term is a negative value, while it is a nonnegative value when the robot is outside the safe region. Once the robot is in the obstacle-free safe region, there are no obstacles between the human and the robot, and the human will see the robot directly. Because human users prefer to keep the service robot in their line of sight for a relative face-to-face interaction, it is a suitable metric to evaluate the possibility of the issue which detour direction will be chosen. The smaller the cost is, the more possible the human route is.

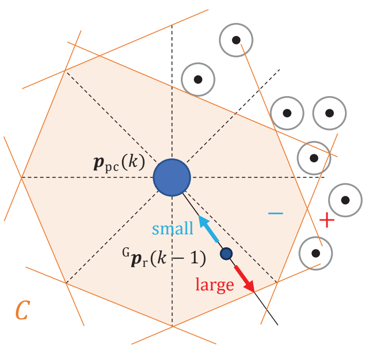

Fig. 1. Diagram of the similarity cost J_{tvl}

D. Implementation Details of the Safe Corridor-Based Model Predictive Control

For the SCMPC in this paper, we give the following costs and constraints:

1) Following Costs:

$$J_{\mathrm{q}}=\sum_{k=k_0+1}^{k_0+\mathrm{N}}\mathrm{k}_{\mathrm{pf}}\left\|{}^{\mathrm{G}}\mathbf{e}_{\mathrm{pf}}(k)\right\|^2+\mathrm{k}_{\mathrm{pl}}\left\|{}^{\mathrm{G}}\mathbf{e}_{\mathrm{pl}}(k)\right\|^2+\mathrm{k}_{\mathrm{vf}}\left\|{}^{\mathrm{G}}\mathbf{e}_{\mathrm{vf}}(k)\right\|^2+\mathrm{k}_{\mathrm{vl}}\left\|{}^{\mathrm{G}}\mathbf{e}_{\mathrm{vl}}(k)\right\|^2, \tag{2a}$$

$$J_{\mathrm{u}}=\sum_{k=k_0}^{k_0+\mathrm{N}}\mathrm{k}_{\mathrm{af}}\left\|{}^{\mathrm{G}}\mathbf{e}_{\mathrm{af}}(k)\right\|^2+\mathrm{k}_{\mathrm{al}}\left\|{}^{\mathrm{G}}\mathbf{e}_{\mathrm{al}}(k)\right\|^2. \tag{2b}$$

For this cost, the position, velocity and acceleration errors are decomposed into the forward and lateral components, and each one is given an independent weight. In detail, ${}^{\mathrm{G}}\mathbf{e}_{\mathrm{pf}}(k)={}^{\mathrm{G}}\mathbf{v}_{\mathrm{rd}}(k)\,{}^{\mathrm{G}}\mathbf{v}_{\mathrm{rd}}^{T}(k)\,{}^{\mathrm{G}}\mathbf{e}_{\mathrm{p}}(k)/\left\|{}^{\mathrm{G}}\mathbf{v}_{\mathrm{rd}}(k)\right\|^2$ is the position error along the forward direction at time k, i.e. the direction of the reference velocity ${}^{\mathrm{G}}\mathbf{v}_{\mathrm{rd}}(k)$, and ${}^{\mathrm{G}}\mathbf{e}_{\mathrm{pl}}(k)={}^{\mathrm{G}}\mathbf{e}_{\mathrm{p}}(k)-{}^{\mathrm{G}}\mathbf{e}_{\mathrm{pf}}(k)$ is the position error along the lateral direction at time k, which is vertical to ${}^{\mathrm{G}}\mathbf{e}_{\mathrm{pf}}(k)$. ${}^{\mathrm{G}}\mathbf{e}_{\mathrm{vf}}(k)$, ${}^{\mathrm{G}}\mathbf{e}_{\mathrm{vl}}(k)$, ${}^{\mathrm{G}}\mathbf{e}_{\mathrm{af}}(k)$ and ${}^{\mathrm{G}}\mathbf{e}_{\mathrm{al}}(k)$ are named and calculated in a similar fashion. The details of the calculation are shown in Fig.2.

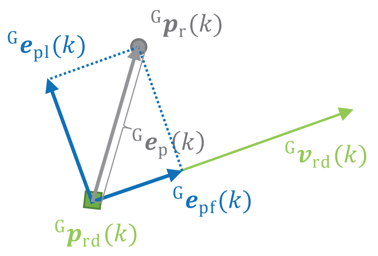

Fig. 2. Diagram of the forward and lateral component of the position error

2) Obstacle Avoidance Cost:

$$J_{\mathrm{obs}}=\mathrm{k}_{\mathrm{obs}}\sum_{k=k_0+1}^{k_0+\mathrm{N}}\sum_{\mathbf{p}\in{}^{\mathrm{G}}P_{\mathrm{obs}}}\frac{\max\left\{\mathrm{d}_{\mathrm{vis}}-\left\|\mathbf{p}-{}^{\mathrm{G}}\mathbf{p}_{\mathrm{r}}(k)\right\|,0\right\}^2}{N_{\mathrm{vis}}(k)}, \tag{3a}$$

$$N_{\mathrm{vis}}(k)=\sum_{\mathbf{p}\in{}^{\mathrm{G}}P_{\mathrm{obs}}}\mathbb{1}\left(\left\|\mathbf{p}-{}^{\mathrm{G}}\mathbf{p}_{\mathrm{r}}(k)\right\|<\mathrm{d}_{\mathrm{vis}}\right), \tag{3b}$$

3) Field of View Cost:

$$J_{\mathrm{vis}}=\sum_{k=k_0+1}^{k_0+\mathrm{N}}\mathrm{k}_{\mathrm{h}}\rho_{\mathrm{h}}^2(k), \tag{4a}$$

$$\rho_{\mathrm{h}}(k)=\left|\theta_{\mathrm{hr}}(k)\right|d_{\mathrm{hr}}(k), \tag{4b}$$

where $d_{\mathrm{hr}}(k)=\left\|{}^{\mathrm{G}}\mathbf{p}_{\mathrm{r}}(k)-{}^{\mathrm{G}}\mathbf{p}_{\mathrm{h}}(k)\right\|$ is the human-robot distance, and $\theta_{\mathrm{hr}}(k)$ is the angle between the bisector of the robot's field of view and human robot connecting line which can be seen in Fig.3.

4) Following Motion Constraints:

$$\begin{cases}\left|{}^{\mathrm{G}}v_{\mathrm{x,r}}(k)\right|<\mathrm{v}_{\max},\ \left|{}^{\mathrm{G}}v_{\mathrm{y.r}}(k)\right|<\mathrm{v}_{\max},\ \left|{}^{\mathrm{G}}\omega_{\mathrm{r}}(k)\right|<\omega_{\max}\\ \left|{}^{\mathrm{G}}a_{\mathrm{x,r}}(k)\right|<\mathrm{a}_{\max},\ \left|{}^{\mathrm{G}}a_{\mathrm{y,r}}(k)\right|<\mathrm{a}_{\max},\ \left|{}^{\mathrm{G}}\beta_{\mathrm{r}}(k)\right|<\beta_{\max}\end{cases}, \tag{5}$$

5) Field of View Constraints:

$$\left|\theta_{\mathrm{hr}}(k)\right|<\theta_{\mathrm{v}},\quad \mathrm{d}_{\mathrm{l}}<d_{\mathrm{hr}}(k)<\mathrm{d}_{\mathrm{u}}, \tag{6}$$

6) Safe Corridor Constraints:

$$\mathbf{A}_{\mathrm{cor}}(k)\,{}^{\mathrm{G}}\mathbf{p}_{\mathrm{r}}(k)-\mathbf{b}_{\mathrm{cor}}(k)\leq-\epsilon_{\mathrm{obs}}, \tag{7}$$

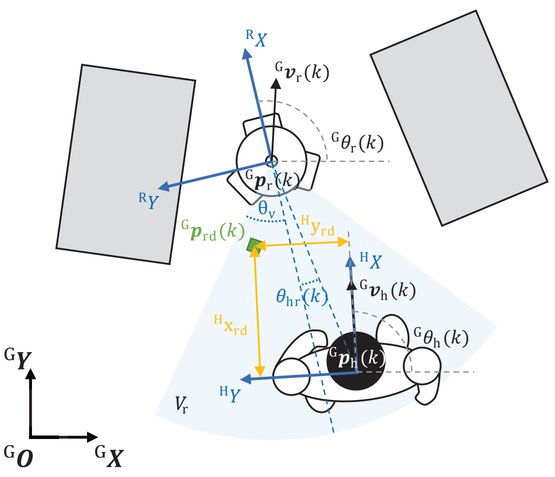

Fig. 3. Overview of human-robot system

To remove the aforementioned inequation constraints, we use a quadratic penalty function. For a constraint in the formation of $a\leq b$, the penalty function is defined as:

$$\max\left\{a-b,0\right\}^2. \tag{8}$$

Furthermore, given the uncertainty of the long-term prediction, the weights of the constraints are first multiplied by a general coefficient k_{c}, and then are divided by a factor λ sequentially as the prediction goes forward. Therefore, the three aforementioned constraints are transformed into the following penalties:

7) Following Motion Penalties:

$$\begin{aligned} J_{\dot{\mathrm{q}},\mathrm{c}}=&\sum_{k=k_0+1}^{k_0+\mathrm{N}}\mathrm{k}_{\mathrm{v,c}}\mathrm{k}_{\mathrm{c}}\left\|\max\left\{\left|{}^{\mathrm{G}}\mathbf{v}_{\mathrm{r}}(\mathbf{k})\right|-\mathrm{v}_{\max},0\right\}\right\|^2/\lambda^{k-k_0}\\ &+\mathrm{k}_{\omega,\mathrm{c}}\mathrm{k}_{\mathrm{c}}\max\left\{\left|{}^{\mathrm{G}}\omega_{\mathrm{r}}(k)\right|-\omega_{\max},0\right\}^2/\lambda^{k-k_0}, \end{aligned} \tag{9a}$$

$$\begin{aligned} J_{\mathrm{u,c}}=&\sum_{k=k_0}^{k_0+\mathrm{N}}\mathrm{k}_{\mathrm{a,c}}\mathrm{k}_{\mathrm{c}}\left\|\max\left\{\left|{}^{\mathrm{G}}\mathbf{a}_{\mathrm{r}}(\mathbf{k})\right|-\mathrm{a}_{\max},0\right\}\right\|^2/\lambda^{k-k_0}\\ &+\mathrm{k}_{\beta,\mathrm{c}}\mathrm{k}_{\mathrm{c}}\max\left\{\left|{}^{\mathrm{G}}\beta_{\mathrm{r}}(k)\right|-\beta_{\max},0\right\}^2/\lambda^{k-k_0}, \end{aligned} \tag{9b}$$

8) Field of View Penalty:

$$\begin{aligned} J_{\mathrm{fov,c}}=&\sum_{k=k_0+1}^{k_0+\mathrm{N}}\mathrm{k}_{\theta,\mathrm{c}}\mathrm{k}_{\mathrm{c}}\max\left\{\left|\theta_{\mathrm{hr}}(k)\right|-\theta_{\mathrm{v}},0\right\}^2/\lambda^{k-k_0}\\ &+\mathrm{k}_{\mathrm{d,c}}\mathrm{k}_{\mathrm{c}}\max\left\{d_{\mathrm{hr}}(k)-\mathrm{d}_{\mathrm{u}},0\right\}^2/\lambda^{k-k_0}\\ &+\mathrm{k}_{\mathrm{d,c}}\mathrm{k}_{\mathrm{c}}\max\left\{\mathrm{d}_{\mathrm{l}}-d_{\mathrm{hr}}(k),0\right\}^2/\lambda^{k-k_0}, \end{aligned} \tag{10}$$

9) Safe Corridor Penalty:

$$J_{\mathrm{cor,c}}=\sum_{k=k_0+1}^{k_0+\mathrm{N}}\mathrm{k}_{\mathrm{cor,c}}\mathrm{k}_{\mathrm{c}}\left\|\max\left\{\mathbf{A}_{\mathrm{cor}}(k)\,{}^{\mathrm{G}}\mathbf{p}_{\mathrm{r}}(k)-\mathbf{b}_{\mathrm{cor}}(k)+\epsilon_{\mathrm{obs}},0\right\}\right\|^2/\lambda^{k-k_0}, \tag{11}$$

Overall, the finite-horizon optimal control problem for the SCMPC is defined as:

$$\begin{aligned} &\underset{{}^{\mathrm{G}}\mathbf{u}_{\mathrm{r,N}}(k_0)}{\arg\min}\ J_{\mathrm{q}}+J_{\mathrm{u}}+J_{\mathrm{obs}}+J_{\mathrm{vis}}+J_{\dot{\mathrm{q}},\mathrm{c}}+J_{\mathrm{u,c}}+J_{\mathrm{fov,c}}+J_{\mathrm{cor,c}}\\ &\text{s.t.}\ {}^{\mathrm{G}}\mathbf{q}_{\mathrm{r}}(k_0)={}^{\mathrm{G}}\mathbf{q}_{\mathrm{r0}},\ {}^{\mathrm{G}}\dot{\mathbf{q}}_{\mathrm{r}}(k_0)={}^{\mathrm{G}}\dot{\mathbf{q}}_{\mathrm{r0}},\\ &\quad\begin{bmatrix}{}^{\mathrm{G}}\mathbf{q}_{\mathrm{r}}(k+1)\\ {}^{\mathrm{G}}\dot{\mathbf{q}}_{\mathrm{r}}(k+1)\end{bmatrix}=A_{\mathrm{r}}\begin{bmatrix}{}^{\mathrm{G}}\mathbf{q}_{\mathrm{r}}(k)\\ {}^{\mathrm{G}}\dot{\mathbf{q}}_{\mathrm{r}}(k)\end{bmatrix}+B_{\mathrm{r}}\,{}^{\mathrm{G}}\mathbf{u}_{\mathrm{r}}(k),\ k=k_0,k_0+1,\ldots,k_0+\mathrm{N}-1. \end{aligned} \tag{12}$$

After substituting the initial states and the kinematics of the robot into the cost functions, the problem is transformed into an unconstrained optimization formulation. Finally, this optimization problem is solved by the BFGS method.