Collision-Aware Velocity Shaping (2D Mockup)

roballgame

State and Geometry

We model a 2D point mass (dot) of radius r at position $p \in \mathbb{R}^2$. The user provides a joystick velocity (or acceleration) command $v_{cmd} \in \mathbb{R}^2$. Obstacle i yields a signed distance d_i and outward unit normal n_i at the closest point. We use the nearest obstacle (d, n) unless stated otherwise.

Circle obstacle. For a circle centered at c with radius R:

$$d = \|p - c\| - (R + r), \qquad n = \frac{p - c}{\|p - c\|}.$$

Axis-aligned walls. For boundaries: left, right, top, bottom.

$$d_{left} = p_x - r, \quad n_{left} = (1, 0)$$

$$d_{right} = (W - p_x) - r, \quad n_{right} = (-1, 0)$$
$$d_{top} = p_y - r, \quad n_{top} = (0, 1)$$
$$d_{bottom} = (H - p_y) - r, \quad n_{bottom} = (0, -1)$$

Axis-aligned rectangles. For a rectangle with top-left (x, y) and size (w, h), expand by r and compute the distance to the nearest point on the expanded rectangle. Let $L = x - r$, $R = x + w + r$, $T = y - r$, $B = y + h + r$ and clamp $q = (p_x, p_y)$ into $\tilde{q} = (\text{clamp}(p_x, L, R), \text{clamp}(p_y, T, B))$. Then $d = \|q - \tilde{q}\|$ and $n = (q - \tilde{q})/\|q - \tilde{q}\|$. If q is inside the expanded rectangle, d is negative and n points to the nearest side.

Line segments. For a segment from a to b, project p onto the segment:

$$t = \text{clamp}\left(\frac{(p - a) \cdot (b - a)}{\|b - a\|^2}, 0, 1\right), \quad c = a + t(b - a)$$

$$d = \|p - c\| - r, \quad n = \frac{p - c}{\|p - c\|}.$$

Model 1: Speed Scaling

Define a scale factor $s(d)$ that shrinks as the distance to obstacles decreases (Figure 1).

$$s(d) = \text{clamp}\left(\frac{d - d_{stop}}{d_{slow} - d_{stop}}, 0, 1\right)$$

$$v_{safe} = s(d)\, v_{cmd}.$$

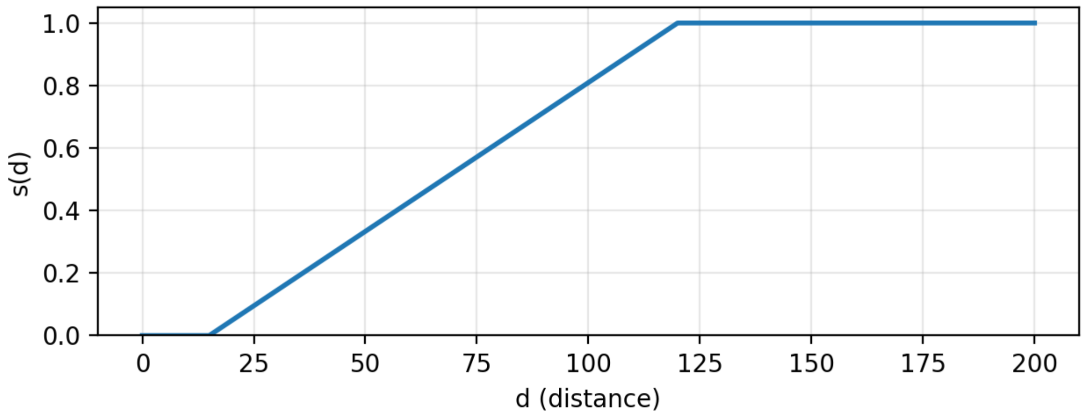

Figure 1: Speed scaling factor used in Models 1 and 4.

Model 2: Repulsive Field

Compute a repulsive velocity from nearby obstacles and add it to the command (Figure 2). For each obstacle with $d_i < d_0$:

$$v_{rep,i} = k\left(\frac{1}{d_i} - \frac{1}{d_0}\right)\frac{1}{d_i^2}\, n_i,$$

and $v_{rep,i} = 0$ for $d_i \geq d_0$. Optionally clamp $\|v_{rep,i}\|$.

$$v_{safe} = v_{cmd} + \sum_i v_{rep,i}.$$

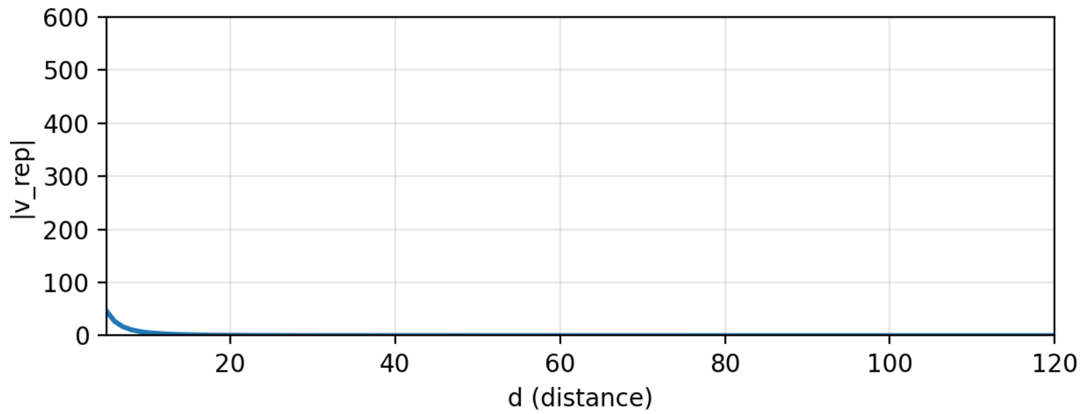

Figure 2: Repulsive field magnitude for Model 2 (example parameters).

Model 3: Normal Projection

Project out motion that pushes into the nearest obstacle surface (Figure 3).

$$v_n = (v_{cmd} \cdot n)\, n$$

If v_{cmd} points into the obstacle, remove that component:

$$v_{safe} = v_{cmd} - \max(0, -v_{cmd} \cdot n)\,(-n)$$

Equivalently, if $v_{cmd} \cdot n < 0$, then

$$v_{safe} = v_{cmd} + (v_{cmd} \cdot n)\,n.$$

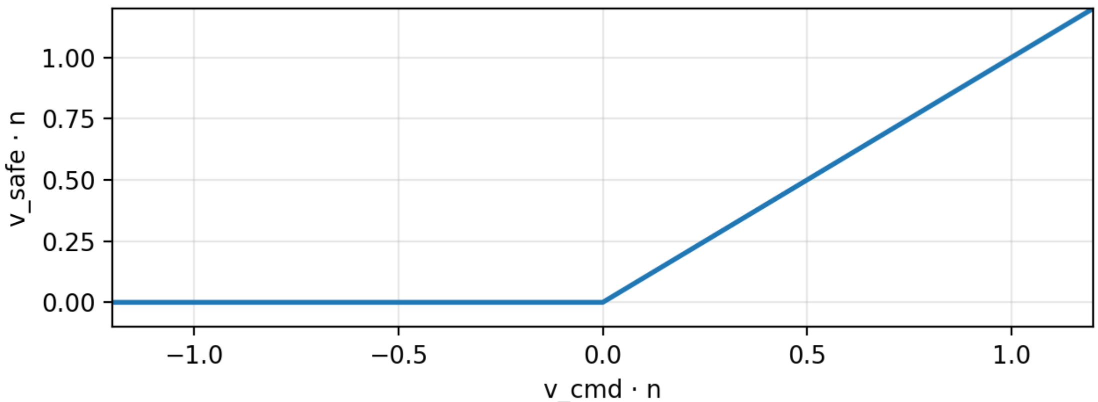

Figure 3: Normal projection (Model 3): motion into the surface is removed.

Model 4: Damped Barrier

Decompose command into normal and tangential components, then damp only the normal component (Figure 4).

$$v_n = (v_{cmd} \cdot n)\,n, \quad v_t = v_{cmd} - v_n$$

$$v_{safe} = s(d)\,v_n + v_t$$

with the same $s(d)$ as Model 1.

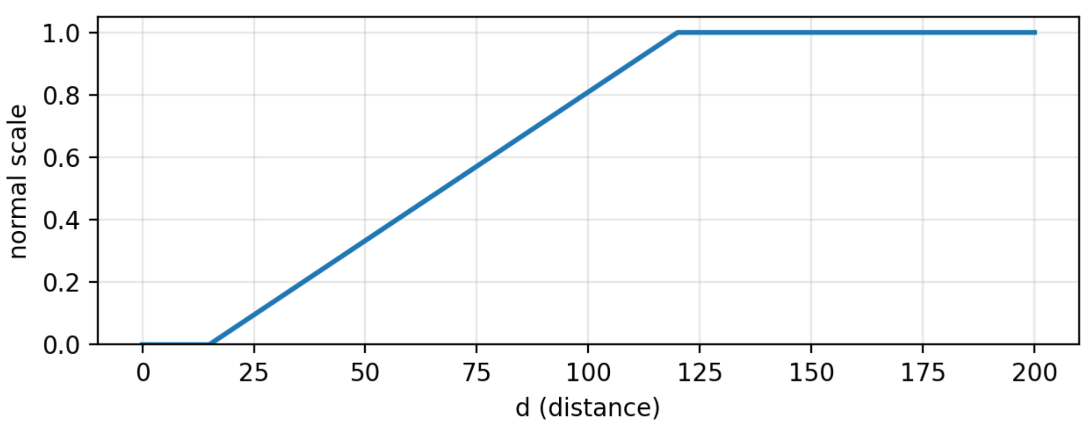

Figure 4: Normal component scaling in the damped barrier (Model 4).

Logging

In the mockup, we log $(t, d, \|v\|, \|v_{cmd}\|)$ each frame to enable comparison of how each model shapes velocity vs. distance.