1 Constants

Symbol	Unit	Value	Explanation
Δt	s	$0.01 - 0.001$	Timestep
τ_{adj}	s	0.5	Characteristic time in which agent adjusts its movement.
k	N	1.5	Social force scaling constant.
τ_0	s	3.0	Interaction time horizon.
μ	kg/s^2	$1.2e+05$	Compression counteraction constant.
κ	kg/(m s)	$2.4e+05$	Sliding friction constant.
A	N	$2.0e+03$	Scaling coefficient for social force.
B	m	0.08	Coefficient for social force.
$\Vert\mathbf{f}_{max}\Vert$	N		Force magnitude limit.

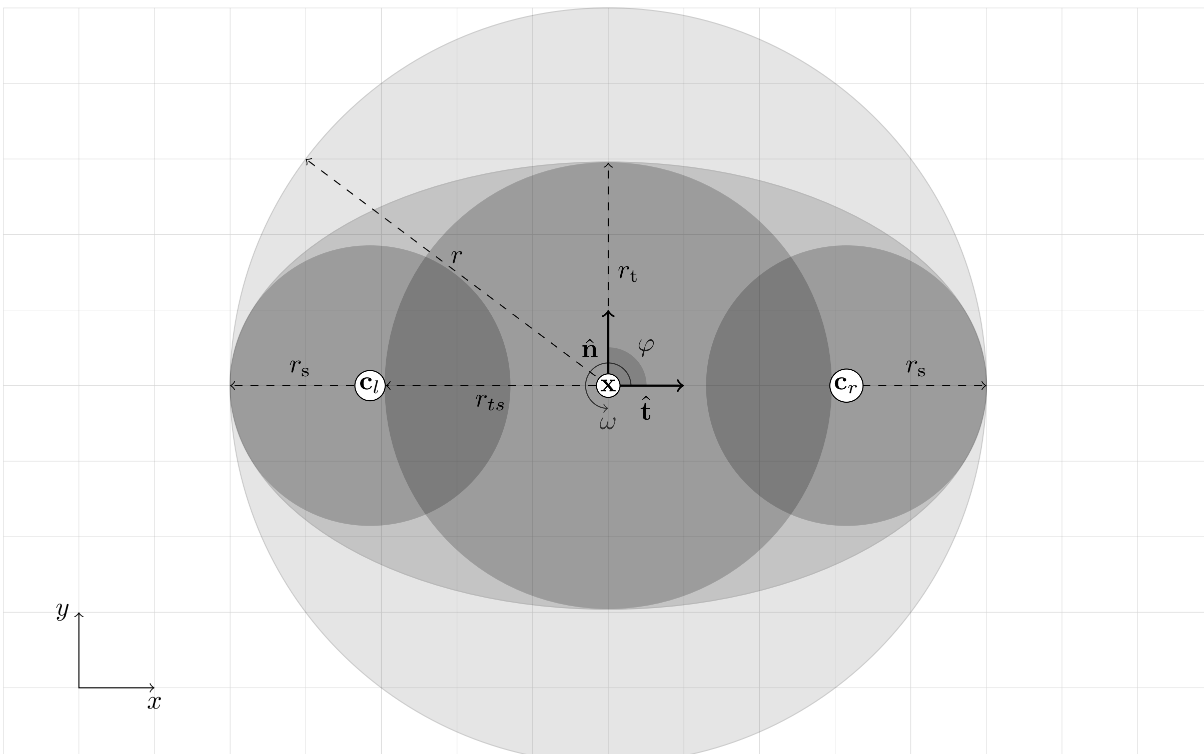

Figure 1: Circle, ellipse and three circle representations of an agent.

2 Agents

2.1 Properties

Table 1: Shoulder, torso and total radii.

	Total		Torso	Shoulder	
	r	$\pm$	r_t/r	r_s/r	r_{ts}/r
adult	0.255	0.035	0.5882	0.3725	0.6275
child	0.210	0.015	0.5714	0.3333	0.6667
eldery	0.250	0.020	0.6000	0.3600	0.6400
female	0.240	0.020	0.5833	0.3750	0.6250
male	0.270	0.020	0.5926	0.3704	0.6296

Table 2: Properties

r	m		Total radius
m	kg	80	Mass
I	$\mathrm{kg} \cdot \mathrm{m}^2$	4.0	Rotational moment
$\mathbf{x}$	m		Position
$\mathbf{v}$	m/s		Velocity
v_0	m/s		Goal velocity
$\hat{\mathbf{e}}_0$			Goal direction
$\hat{\mathbf{e}}$			Target direction
φ	rad	$[0, 2\pi]$	Body angle
ω	rad/s		Angular velocity
φ_0	rad		Target angle
ω_0	rad/s	4π	Max angular velocity
p		$0-1$	Herding tendency

Table 3: Relative

$\tilde{\mathbf{x}} = \mathbf{x}_i - \mathbf{x}_j$	Relative position
$\tilde{\mathbf{v}} = \mathbf{v}_i - \mathbf{v}_j$	Relative velocity

2.2 Circular agent

$d = \lVert\tilde{\mathbf{x}}\rVert$	Distance
$\hat{\mathbf{n}} = \tilde{\mathbf{x}}/d$	Normal vector
$\hat{\mathbf{t}} = R(-90°) \cdot \hat{\mathbf{n}}$	Tangent vector

Total radius and relative distance

$$\tilde{r} = r_i + r_j$$
$$h = \tilde{r} - d$$

2.3 Three circles

$$\mathbf{c}_r = \hat{\mathbf{t}} r_{ts} + \mathbf{x}$$
$$\mathbf{c}_l = -\hat{\mathbf{t}} r_{ts} + \mathbf{x}$$
$$\hat{\mathbf{t}} = \begin{bmatrix} -\sin(\varphi), \cos(\varphi) \end{bmatrix}$$

$$\mathbf{h} = \left(\begin{bmatrix} r_t & r_s & r_s \end{bmatrix}_i + \begin{bmatrix} r_t \\ r_s \\ r_s \end{bmatrix}_j \right) - \left\lVert \begin{bmatrix} \mathbf{x} & \mathbf{c}_r & \mathbf{c}_l \end{bmatrix}_i - \begin{bmatrix} \mathbf{x} \\ \mathbf{c}_r \\ \mathbf{c}_l \end{bmatrix}_j \right\rVert$$

$$\left\lVert \begin{bmatrix} 0 & \hat{\mathbf{t}} r_{ts} & -\hat{\mathbf{t}} r_{ts} \end{bmatrix}_i - \begin{bmatrix} 0 \\ \hat{\mathbf{t}} r_{ts} \\ -\hat{\mathbf{t}} r_{ts} \end{bmatrix}_j + (\mathbf{x}_i - \mathbf{x}_j) \right\rVert$$

$$\left\lVert \begin{bmatrix} 0 & 1 & -1 \end{bmatrix} \left(\hat{\mathbf{t}} r_{ts}\right)_i - \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix} \left(\hat{\mathbf{t}} r_{ts}\right)_j + \tilde{\mathbf{x}} \right\rVert$$

1. Find $h_{max} = \max(\mathbf{h})$ and keep tract of the circles that maximizes this distance.
2. Two circles found $(\mathbf{p}_0, r_0)$ and $(\mathbf{p}_1, r_1)$
3. Compute $\mathbf{r}_i^{soc}$
4. If $h > 0$ compute $\mathbf{r}_i^c$ else $\mathbf{0}$
5. Return $(h_{max}, \mathbf{r}_i^{soc}, \mathbf{r}_i^c)$

2.4 Rotational equation

Rotational equation of motion

$$I \frac{d^2}{dt^2} \varphi(t) = M(t) + \eta(t),$$

where $\eta(t)$ is small random fluctuation torque, and $M(t)$ is total torque, which is the sum of contact, social and motivational torque

$$M_i(t) = M_i^c + M_i^{soc} + M_i^{\tau}$$

Torque from contact forces

$$\mathbf{M}_i^c = \sum_{j \neq i} \left(\mathbf{r}_i^c \times \mathbf{f}_{ij}^c \right)$$

and from social forces

$$\mathbf{M}_i^{soc} = \sum_{j \neq i} \left(\mathbf{r}_i^{soc} \times \mathbf{f}_{ij}^{soc} \right)$$

Motivational torque

$$M_i^\tau = \frac{I_i}{\tau_i}\big((\varphi_i(t) - \varphi_i^0)\omega^0 - \omega(t)\big)$$

As vector

$$\mathbf{r} \times \mathbf{f} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ R_1 & R_2 & R_3 \\ f_1 & f_2 & f_3 \end{vmatrix}$$

In two dimensions

$$\begin{vmatrix} R_1 & R_2 \\ f_1 & f_2 \end{vmatrix} = R_1 \cdot f_2 - R_2 \cdot f_1$$

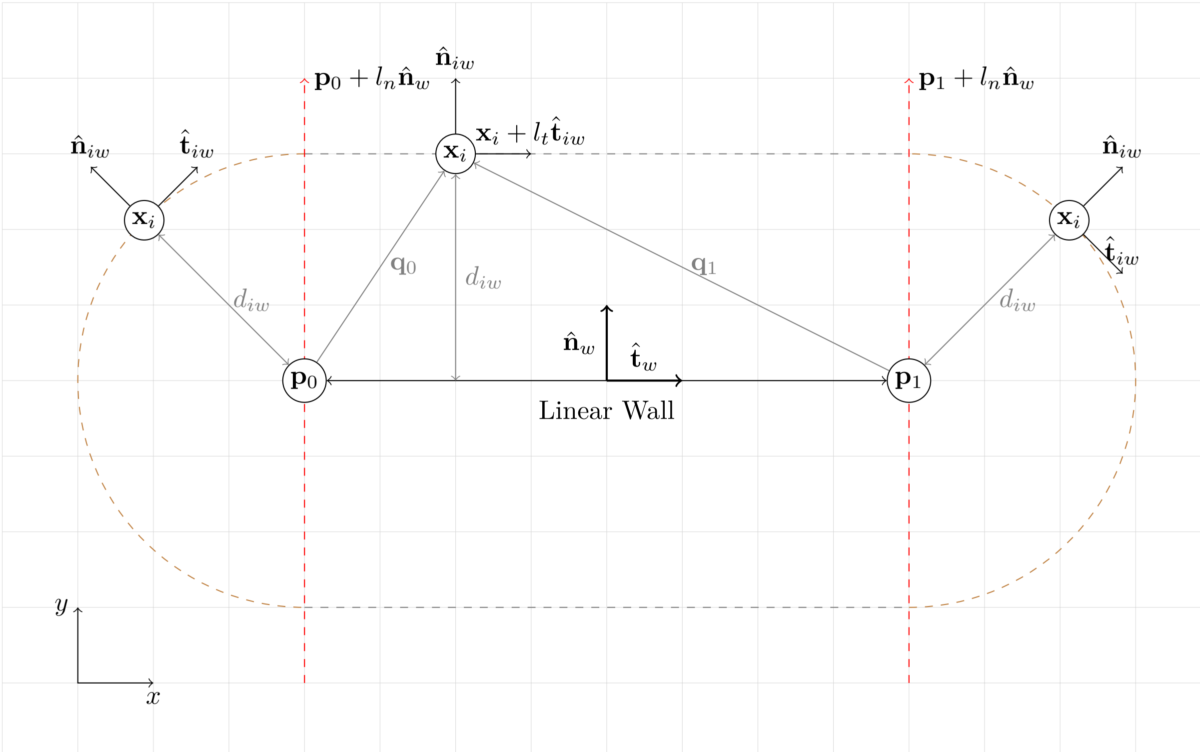

Figure 2: Absolute distance from a linear wall.

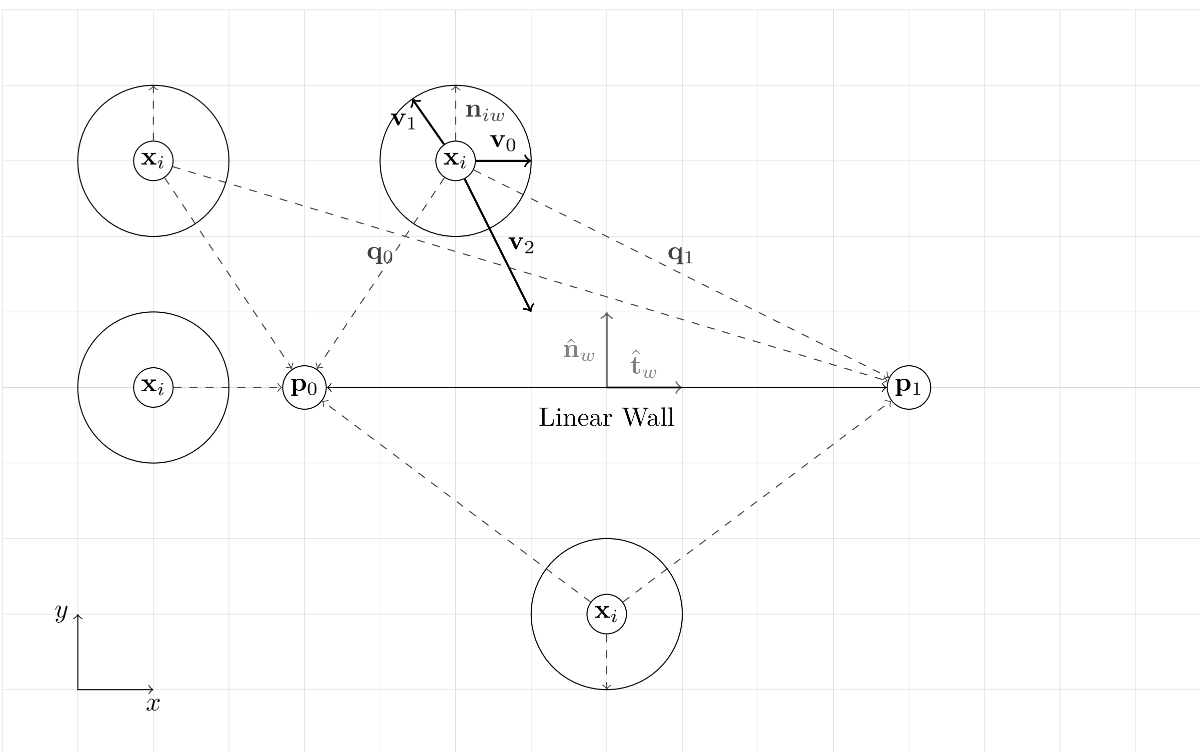

Figure 3: Velocity dependent distance from a linear wall.

3 Linear wall

3.1 Properties

$\mathbf{p}_0$	Start point
$\mathbf{p}_1$	End point
$h_{iw} = r_i - d_{iw}$	
$l_w = \|\mathbf{p}_1 - \mathbf{p}_0\|$	Length
$\hat{\mathbf{t}}_w = (\mathbf{p}_1 - \mathbf{p}_0)/l_w$	
$\hat{\mathbf{n}}_w = R(90°) \cdot \hat{\mathbf{t}}_w$	

3.2 Absolute distance

Solving linear system of equations determining the position of the agent $\mathbf{x}_i$ in relation to wall

$$\begin{cases} \mathbf{p}_0 + l_{n_0}\hat{\mathbf{n}}_w = \mathbf{x}_i + l_{t_0}\hat{\mathbf{t}}_w \\ \mathbf{p}_1 + l_{n_1}\hat{\mathbf{n}}_w = \mathbf{x}_i + l_{t_1}\hat{\mathbf{t}}_w \end{cases}$$

$$\begin{cases} l_{n_0}\hat{\mathbf{n}}_w - l_{t_0}\hat{\mathbf{t}}_w = \mathbf{x}_i - \mathbf{p}_0 = \mathbf{q}_0 \\ l_{n_1}\hat{\mathbf{n}}_w - l_{t_1}\hat{\mathbf{t}}_w = \mathbf{x}_i - \mathbf{p}_1 = \mathbf{q}_1 \end{cases}$$

In matrix form

$$\begin{bmatrix} l_{n_0} & l_{n_1} \\ l_{t_0} & l_{t_1} \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} \mathbf{q}_0 & \mathbf{q}_1 \end{bmatrix}$$

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} \hat{\mathbf{n}}_w & -\hat{\mathbf{t}}_w \end{bmatrix} = \begin{bmatrix} -t_1 & -t_0 \\ t_0 & -t_1 \end{bmatrix} \\ \mathbf{A}^{-1} &= \frac{1}{t_0^2 + t_1^2} \begin{bmatrix} -t_1 & t_0 \\ -t_0 & -t_1 \end{bmatrix} = \begin{bmatrix} -t_1 & t_0 \\ -t_0 & -t_1 \end{bmatrix} \\ &= \begin{bmatrix} \hat{\mathbf{n}}_w \\ -\hat{\mathbf{t}}_w \end{bmatrix} = \mathbf{A}^T \end{aligned}$$

Conditions

$$\begin{aligned} l_n &= l_{n_0} \vee l_{n_1} = \hat{\mathbf{n}}_w \cdot \mathbf{q}_0 \vee \hat{\mathbf{n}}_w \cdot \mathbf{q}_1 \\ l_t &= l_{t_1} + l_{t_0} = -\hat{\mathbf{t}}_w \cdot \mathbf{q}_1 - \hat{\mathbf{t}}_w \cdot \mathbf{q}_0 \end{aligned}$$

Distance between agent and linear wall

$$d_{iw} = \begin{cases} \|\mathbf{q}_0\| & l_t > l_w \\ |l_n| & \text{otherwise} \\ \|\mathbf{q}_1\| & l_t < -l_w \end{cases}$$

Normal vector away from the wall

$$\hat{\mathbf{n}}_{iw} = \begin{cases} \hat{\mathbf{q}}_0 & l_t > l_w \\ \operatorname{sign}(l_n)\hat{\mathbf{n}}_w & \text{otherwise} \\ \hat{\mathbf{q}}_1 & l_t < -l_w \end{cases}$$

3.3 Velocity relative distance

$\tilde{\mathbf{x}} = \mathbf{x}_{iw}$	Relative position
$\tilde{\mathbf{v}} = \mathbf{v}_{iw} = \mathbf{v}_i$	Relative velocity
$\tilde{r} = r_{iw}$	Total radius
$d = \|\tilde{\mathbf{x}}\|$	Distance
$h = \tilde{r} - d$	Relative distance

Dividing vectors

$$\begin{aligned} \mathbf{q}_0 &= \mathbf{p}_0 - \mathbf{x} \\ \mathbf{q}_1 &= \mathbf{p}_1 - \mathbf{x} \\ \hat{\mathbf{n}}_{iw} &= -\operatorname{sign}(\hat{\mathbf{n}}_w \cdot \mathbf{q}_0)\hat{\mathbf{n}}_w \end{aligned}$$

Angle of 2D vector is found using `https://en.wikipedia.org/wiki/Atan2` where angle is between $[-\pi, \pi]$

$$\boldsymbol{\alpha} = [\operatorname{angle}(\mathbf{q}_0), \operatorname{angle}(\mathbf{q}_1), \operatorname{angle}(\hat{\mathbf{n}}_{iw})]$$

$$\varphi = \operatorname{angle}(\mathbf{v})$$

4 Crowd dynamics

4.1 Social force model

Total force exerted on the agent is the sum of movement adjusting, social and contact forces between other agents and wall.

$$\mathbf{f}_i(t) = \mathbf{f}_i^{adj} + \sum_{j \neq i} \left(\mathbf{f}_{ij}^{soc} + \mathbf{f}_{ij}^{c} \right) + \sum_{w} \left(\mathbf{f}_{iw}^{soc} + \mathbf{f}_{iw}^{c} \right) + \boldsymbol{\xi}_i$$

4.2 Adjusting force

Force adjusting agent's movement towards desired in some characteristic time

$$\mathbf{f}^{adj} = \frac{m}{\tau^{adj}} (v_0 \cdot \hat{\mathbf{e}} - \mathbf{v})$$

4.3 Social force

Psychological force for collision avoidance

4.3.1 Velocity independent

$$\mathbf{f}^{soc} = A \exp\left(\frac{h}{B}\right) \hat{\mathbf{n}}$$

4.3.2 Velocity dependent

$$\begin{aligned}
\mathbf{f}^{soc} &= -\nabla_{\tilde{\mathbf{x}}} E(\tau) \\
&= -\nabla_{\tilde{\mathbf{x}}} \left(\frac{k}{\tau^2} \exp\left(-\frac{\tau}{\tau_0} \right) \right) \\
&= -\left(\frac{k}{a\tau^2} \right) \left(\frac{2}{\tau} + \frac{1}{\tau_0} \right) \exp\left(-\frac{\tau}{\tau_0} \right) \left(\tilde{\mathbf{v}} - \frac{a\tilde{\mathbf{x}} + b\tilde{\mathbf{v}}}{d} \right),
\end{aligned}$$

where

$$\begin{aligned}
a &= \tilde{\mathbf{v}} \cdot \tilde{\mathbf{v}} \\
b &= -\tilde{\mathbf{x}} \cdot \tilde{\mathbf{v}} \\
c &= \tilde{\mathbf{x}} \cdot \tilde{\mathbf{x}} - \tilde{r}^2 \\
d &= \sqrt{b^2 - ac}, \quad b^2 - ac > 0 \\
\tau &= \frac{b - d}{a} > 0.
\end{aligned}$$

4.4 Contact force

Physical contact force

$$\mathbf{f}^c = h \cdot \left(\mu \cdot \hat{\mathbf{n}} - \kappa \cdot (\mathbf{v} \cdot \hat{\mathbf{t}}) \hat{\mathbf{t}} \right), \quad h > 0$$

4.5 Random Fluctuation

Uniformly distributed random fluctuation force

$$\boldsymbol{\xi} = f \cdot \left[\cos(\varphi), \quad \sin(\varphi) \right],$$

where

$$f \in [0, f_{max}], \quad \varphi \in [0, 2\pi)$$

4.6 Target direction

Herding behavior

$$\mathbf{e}_i = (1 - p_i)\hat{\mathbf{e}}_i^0 + p_i \left\langle \hat{\mathbf{e}}_j^0 \right\rangle_i$$

5 Integrators

5.1 Differential systems

Angle and angular velocity

$$I \frac{d^2}{dt^2} \varphi(t) = M(t)$$

Position and velocity

$$m \frac{d^2}{dt^2} \mathbf{x}(t) = \mathbf{f}(t)$$

5.2 Numerical methods

Updating using discrete time step Δt

$$\begin{aligned}
t_0 &= 0 \\
t_1 &= t_0 + \Delta t \\
&\vdots \\
t_k &= t_{k-1} + \Delta t
\end{aligned}$$

5.3 Excplicit Euler Method

Angular acceleration

$$\begin{aligned}
\alpha_k &= M_k / I \\
\omega_{k+1} &= \omega_k + \alpha_k \Delta t \\
\varphi_{k+1} &= \varphi_k + \omega_{k+1} \Delta t
\end{aligned}$$

Acceleration on an agent

$$\begin{aligned}
a_k &= \mathbf{f}_k / m \\
\mathbf{v}_{k+1} &= \mathbf{v}_k + a_k \Delta t \\
\mathbf{x}_{k+1} &= \mathbf{x}_k + \mathbf{v}_{k+1} \Delta t
\end{aligned}$$

5.4 Velocity verlet

Velocity verlet algorithm

$$
\begin{aligned}
\mathbf{v}_{k+\frac{1}{2}} &= \mathbf{v}_k + \frac{1}{2} a_k \Delta t \\
\mathbf{x}_{k+1} &= \mathbf{x}_k + \mathbf{v}_{k+\frac{1}{2}} \Delta t \\
\mathbf{v}_{k+1} &= \mathbf{v}_{k+\frac{1}{2}} + \frac{1}{2} a_{k+1} \Delta t
\end{aligned}
$$

or more simply

$$
\begin{aligned}
\mathbf{x}_{k+1} &= \mathbf{x}_k + \mathbf{v}_k \Delta t + \frac{1}{2} a_k \Delta t^2 \\
\mathbf{v}_{k+1} &= \mathbf{v}_k + \frac{1}{2} \left(a_k + a_{k+1} \right) \Delta t
\end{aligned}
$$

6 Spatial game

6.1 Game matrix

$T_i = \lambda_i / \beta$	Estimated evacuation time
λ_i	Number of other agents closer to the exit
β	Capacity of the exit
$T_{ij} = (T_i + T_j)/2$	Average evacuation time
T_{ASET}	Available safe egress time
T_0	Time difference between T_{ASET} and T_i before agents start playing the game
$C > 0$	Cost of conflict

Cost function

$$u(T_i, T_{ASET}), \quad u'(T_i) \geq 0, \quad u''(T_i) \geq 0$$

Increase/decrease in cost

$$\Delta u(T_{ij}) = u(T_{ij}) - u(T_{ij} - \Delta T) \approxeq u'(T_{ij}) \Delta T$$

	Impatient	Patient
Impatient	C, C	$-\Delta u(T_{ij}), \Delta u(T_{ij})$
Patient	$\Delta u(T_{ij}), -\Delta u(T_{ij})$	$0, 0$

	Impatient	Patient
Impatient	$\frac{C}{\Delta u(T_{ij})}, \frac{C}{\Delta u(T_{ij})}$	$-1, 1$
Patient	$1, -1$	$0, 0$

i) Prisoner's dilemma (PD)

$$0 < \frac{C}{\Delta u(T_{ij})} \leq 1$$

ii) Hawk-dove (HD)

$$\frac{C}{\Delta u(T_{ij})} > 1$$

Assumptions

1. Game is not played

$$T_{ij} \leq T_{ASET} - T_0$$

2. Cost function starts to increase quadratically

$$T_{ij} > T_{ASET} - T_0$$

3. Game turns into prisoner's dilemma

$$u'(T_{ASET}) = C, \quad T_{ij} \geq T_{ASET}$$

Cost function that meets the assumptions

$$u(T_{ij}) = \begin{cases} 0 & T_{ij} \leq T_{ASET} - T_0 \\ \frac{C}{2T_0} \left(T_{ij} - T_{ASET} + T_0 \right)^2 & T_{ij} > T_{ASET} - T_0 \end{cases}$$

Derivative

$$u'(T_{ij}) = \begin{cases} 0 & T_{ij} \leq T_{ASET} - T_0 \\ \frac{C}{T_0} \left(T_{ij} - T_{ASET} + T_0 \right) & T_{ij} > T_{ASET} - T_0 \end{cases}$$

Loss/gain of overtaking

$$\Delta u(T_{ij}) \approxeq u'(T_{ij}) \Delta T = \frac{C}{T_0} \left(T_i - T_{ASET} + T_0 \right) \Delta T$$

Value parameter of the game matrix

$$\frac{C}{\Delta u(T_{ij})} \approxeq \frac{T_0}{T_{ij} - T_{ASET} + T_0}$$

6.2 Settings and best-response dynamics

	Unit	Value	
		4	von Neumann neighborhood
		8	Moore neighborhood
r_n	m	0.40	Distance to agent that is considered as neighbor
v_i			Loss defined by game matrix
S			Set of strategies $\{\text{Patient}, \text{Impatient}\}$
s			Strategy $\in \{\text{Patient}, \text{Impatient}\}$

The best-response strategy

$$s_i^{(t)} = \arg\min_{s_i' \in S} \sum_{j \in N_i} v_i \left(s_i', s_j^{(t-1)}; T_{ij} \right)$$

$s_j^{(t-1)}$ strategy neighbor played on period $t-1$

Updating strategy using poisson process.