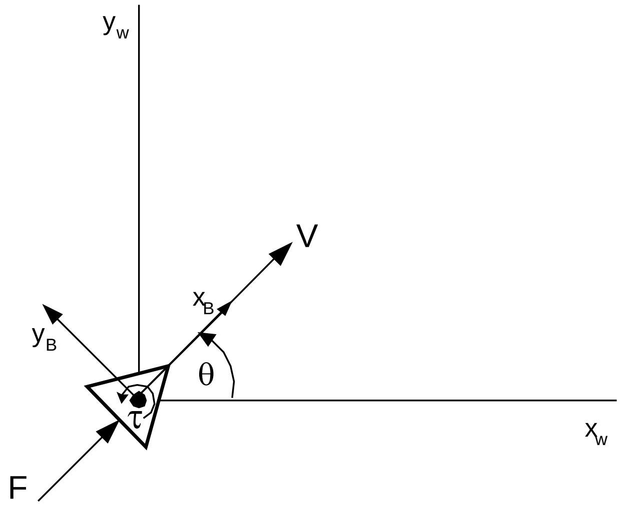
Behavior-based Control Examples

See paper by Craig Reynolds entitled "Steering Behaviors for Autonomous Characters" for more details.

Basic Vehicle Dynamics



Translational Dynamics: $m\dot{V} = F$

Rotational Dynamics: $I\ddot{\theta} = \tau$

The velocity vector **V** of the vehicle can be represented in a coordinate system attached to the body (i.e. body coordinates) or the world coordinate system (i.e. world coordinates)

$\mathbf{V}_w$ = Velocity vector **V** in world coordinates
$\mathbf{V}_B$ = Velocity vector **V** in body coordinates

Relationship between $\mathbf{V}_w$ and $\mathbf{V}_B$:

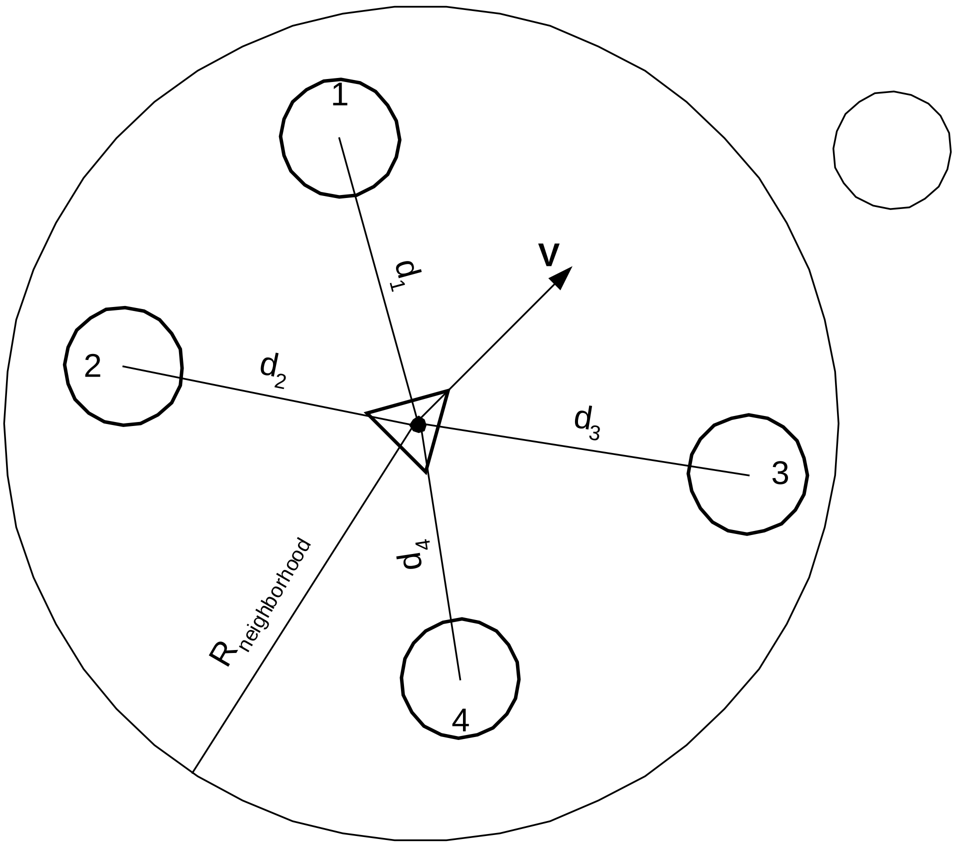
$\mathbf{V}_w = \mathbf{R}(z,\theta)\mathbf{V}_B$ where R is a z-axis rotation matrix

Vehicle Constraints:

F_{max} = max force
τ_{max} = max torque
V_{max} = max velocity
$\dot{\theta}_{max}$ = max angular velocity

Group Behaviors

Separation

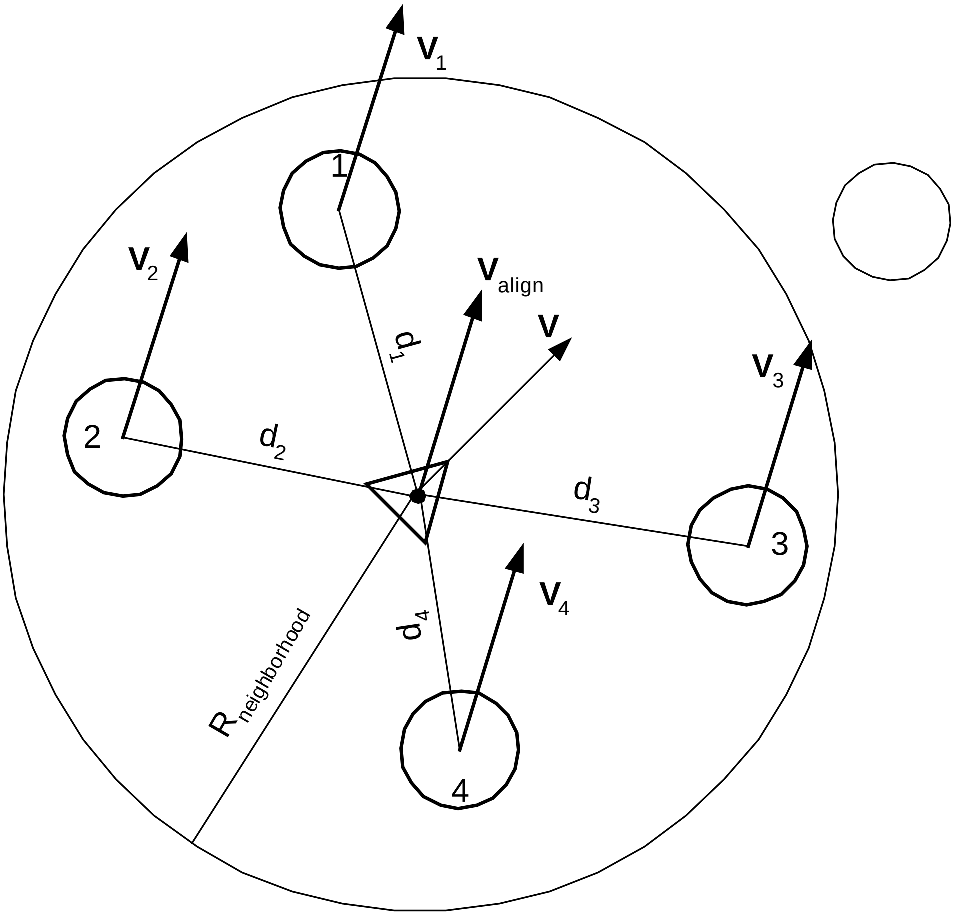


- Determine vehicles and objects in local neighborhood of radius = $R_{neighborhood}$
- Given the distance to each vehicle/object in the neighborhood, compute the separation velocity as follows:

$$\mathbf{V}_{separate} = K_{separate} \sum w_i \frac{\mathbf{d}_i}{\|\mathbf{d}_i\|^2}$$

as the weighted sum of the inverse of the separation distances $\mathbf{d}_i$

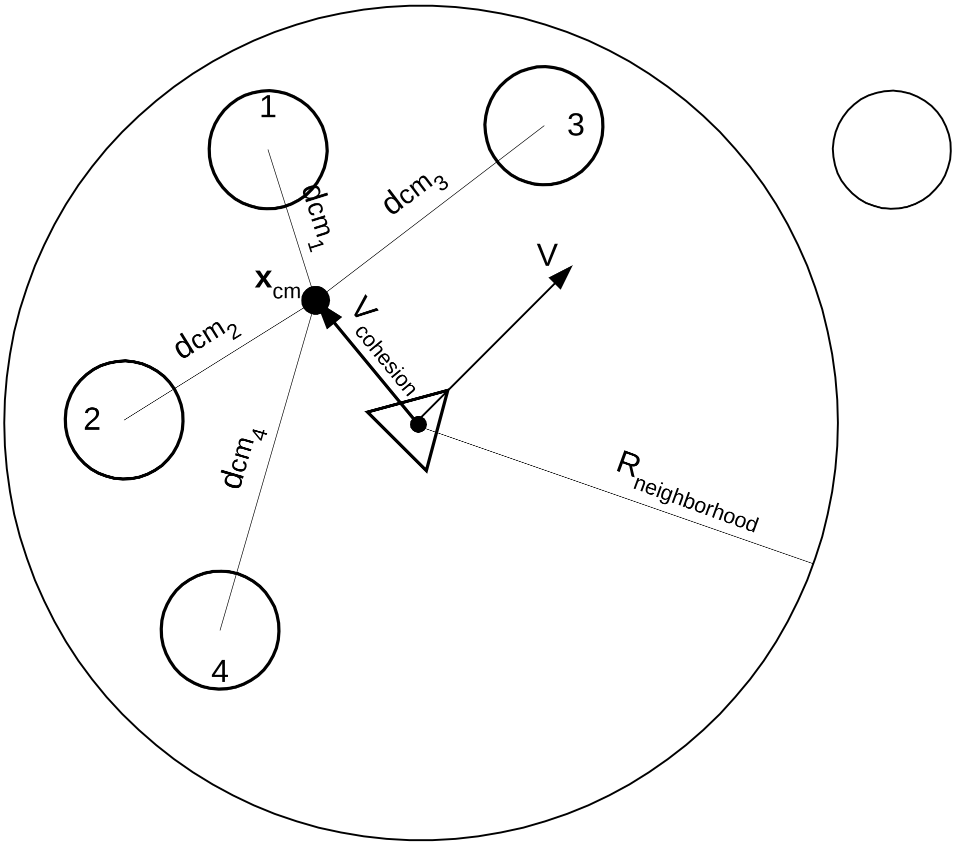
Alignment



- Determine vehicles in local neighborhood of radius = $R_{neighborhood}$
- Given the velocity of each vehicle in the neighborhood, compute the alignment velocity as the weighted average:

$$\mathbf{V}_{align} = K_{align} \frac{\sum w_i \mathbf{V}_i}{\sum w_i}$$

Cohesion



- Determine vehicles in local neighborhood of radius = $R_{neighborhood}$
- Given the position of each vehicle in the neighborhood, $\mathbf{x}_i$, compute the center of mass of the vehicles as the average of their current positions:

$$\mathbf{x}_{cm} = \frac{\sum w_i \mathbf{x}_i}{\sum w_i}$$

- Compute $\mathbf{V}_{cohesion}$ as the vector from the vehicles current position ($\mathbf{x}$) to the center of mass position of all the vehicles in the neighborhood ($\mathbf{x}_{cm}$):

$$\mathbf{V}_{cohesion} = K_{cohesion}(\mathbf{x}_{cm} - \mathbf{x})$$

Flocking

- Flocking behaviors are created as a weighted sum of separation, cohesion and alignment velocity commands

$$\mathbf{V}_{flock} = c_{separate}\mathbf{V}_{separate} + c_{cohesion}\mathbf{V}_{cohesion} + c_{alignment}\mathbf{V}_{alignment}$$

where $c_{separate}$, $c_{cohesion}$ and $c_{alignment}$ are the coefficients which determine the relative weighting of the separation, cohesion and alignment velocity commands

- Control law command variables V_d and θ_d for each vehicle are of the form:

    - $V_d = \|\mathbf{V}_{flock}\|$
    - Given $\mathbf{V}_{flock}$, compute θ_d as in Seek behavior

Leader Following

- Leader Following behaviors are created as a weighted average of separation and arrival commands

$$\mathbf{V}_{leaderfollow} = c_{separate}\mathbf{V}_{separate} + c_{arrival}\mathbf{V}_{arrival}$$

where $c_{separate}$, $c_{arrival}$ are the coefficients which determine the relative weighting of the separation and arrival velocity commands

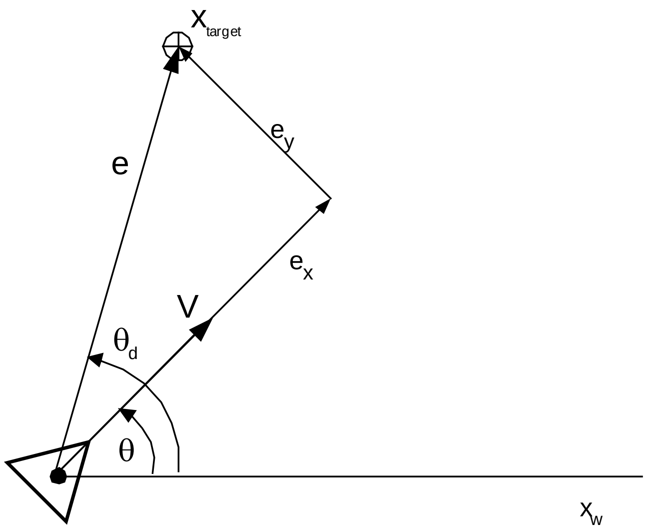
- Compute the target location for the arrival command to be a point offset slightly behind the leader, which can vary based on the vehicle velocity (i.e. tighter or looser formation)

- Control law command variables V_d and θ_d for each vehicle are of the form:

    - $V_d = \|\mathbf{V}_{leaderfollow}\|$
    - Given $\mathbf{V}_{leaderfollow}$, compute θ_d as in Seek behavior

Individual Behaviors

Seek/Flee



- Compute error, $\mathbf{e} = \mathbf{x}_{target} - \mathbf{x}$, between target location and present location
- Compute components of error $\mathbf{e}_x$ and $\mathbf{e}_y$ in body axes
- Compute desired heading angle command (θ_d) for Seek as follows:

  Seek: $\theta_d = \theta + \tan^{-1}(\frac{\mathbf{e_y}}{\mathbf{e_x}})$ where $\theta_d \in [-\pi, \pi]$

- Compute desired heading angle command (θ_d) for Flee as follows:

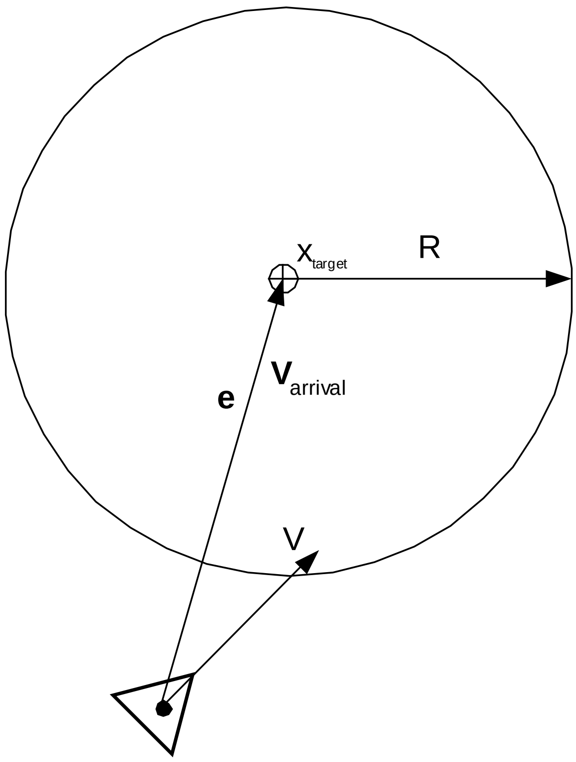
For Flee, the desired direction is opposite that of the error vector. This can be achieved using the arctan function by negating the sign of the error components which yields:

Flee: $\theta_d = \theta + \tan^{-1}(\frac{\mathbf{e_y}}{\mathbf{e_x}}) + \pi$ where $\theta_d \in [-\pi, \pi]$

If using arctan2 can be computed as $\theta_d = \theta + \tan^{-1}(\frac{-\mathbf{e_y}}{-\mathbf{e_x}})$

- For both Seek and Flee, set the desired velocity command $V_d = V_{max}$

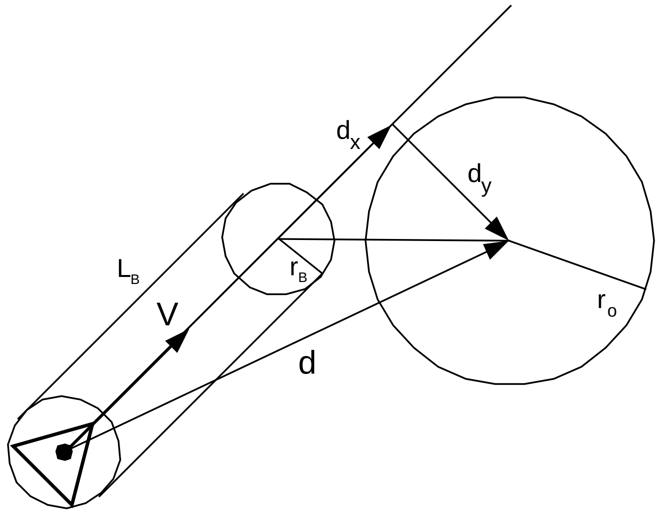
Arrival/Departure



- Arrival
  - Compute $\mathbf{V}_{arrival} = K_{arrival}\ \mathbf{e}$
  - Control law command variables V_d and θ_d are compute as follows:
    - $V_d = \|\mathbf{V}_{\text{arrival}}\|$
    - Given the desired direction of the vector $\mathbf{V}_{\text{arrival}}$, compute θ_d as in Seek behavior
- Departure
  - Compute $\mathbf{V}_{departure} = K_{departure} \dfrac{-\mathbf{e}}{\|\mathbf{e}\|^2}$

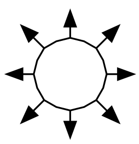
    where $K_{departure}$ is a scalar gain that determines the strength of repulsion
  - Control law command variables V_d and θ_d are compute as follows:
    - $V_d = \|\mathbf{V}_{\text{departure}}\|$
  - Given $\mathbf{V}_{\text{departure}}$, compute θ_d as in Flee behavior

Obstacle Avoidance



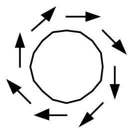
- Approximate vehicle and obstacle using bounding spheres
  - r_B = radius of vehicle bounding sphere
  - r_o = radius of obstacle bounding sphere
- Goal is to keep an imaginary cylinder of free space of length L_B and radius r_B in front of vehicle at all times
- Length of cylinder L_B is computed as a function of current velocity, $L_B = T_{avoid} * \|\mathbf{V}\|$
- Detect potential collision as follows:
  - Compute $\mathbf{d}_{world}$ = position_obstacle – position_vehicle
  - Localize $\mathbf{d}_{world}$ into body coordinates to get d
  - If $\|d_x\| > L_B$, then no collision
  - If $\|d_x\| <= L_B$, then potential collision
    - If $\|d_y\| > r_B + r_o$, then no collision
    - If $\|d_y\| <= r_B + r_o$, then potential for collision.
    - Take appropriate corrective action by computing $\mathbf{V}_{avoid}$ using either a normal or tangential velocity field approach

- Calculation of $\mathbf{V}_{avoid}$
  - ***Normal velocity field*** **–** the vehicle is repulsed from the obstacle in a direction $\hat{n}_{avoid}$, which is a unit vector normal to the obstacle at the point of intersection with **d**.



- Direction of $\mathbf{V}_{\text{avoid}}$, $\hat{\mathbf{n}}_{avoid} = \dfrac{-\mathbf{d}}{\|\mathbf{d}\|}$
- Magnitude of $\mathbf{V}_{\text{avoid}}$, $\|\mathbf{V}_{avoid}\| = \dfrac{K_{aviod}V_{\max}}{1+(\|\mathbf{d}\|-(r_B+r_o))^2}$ where K_{avoid} is a scalar gain that determines the strength of repulsion

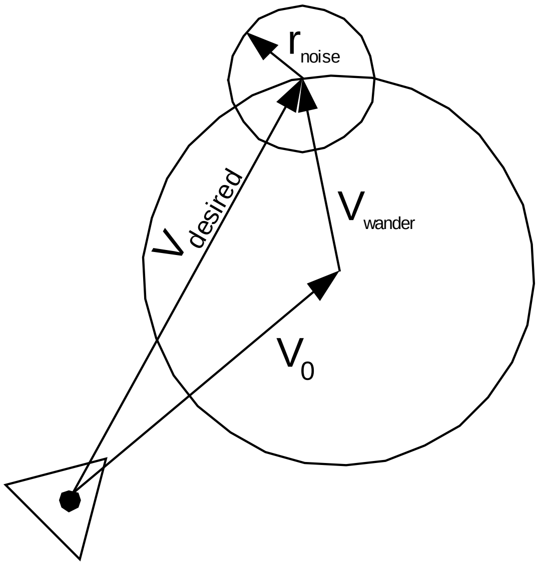
  - ***Tangential velocity field–*** the vehicle is repulsed from the obstacle in a direction $\hat{t}_{avoid}$, which is a unit vector tangent to the obstacle at the point of intersection with d.



    - Direction of $\mathbf{V}_{\text{avoid}}$, $\hat{t}_{avoid}$, can be found as follows:
        - $\hat{\mathbf{n}}_{avoid} = \dfrac{-\mathbf{d}}{\|\mathbf{d}\|}$
        - $\hat{\mathbf{k}}_{avoid} = \dfrac{\mathbf{V}\times\mathbf{d}}{\|\mathbf{V}\|\|\mathbf{d}\|}$
        - $\hat{\mathbf{t}}_{avoid} = \hat{\mathbf{k}}_{avoid} \times \hat{\mathbf{n}}_{avoid}$
    - Magnitude of $\mathbf{V}_{\text{avoid}}$, $\|\mathbf{V}_{avoid}\| = \dfrac{K_{aviod}V_{\max}}{1+(\|\mathbf{d}\|-(r_B+r_o))^2}$ where K_{avoid} is a scalar gain that determines the strength of repulsion

- Control law command variables V_d and θ_d are compute as follows:
  - $V_d = \|\mathbf{V}_{\text{avoid}}\|$
  - Given $\mathbf{V}_{\text{avoid}}$, compute θ_d as in Seek behavior

Wander



- Wandering involves doing a random walk on a circle of radius $\|\mathbf{V}_{wander}\|$ centered at $\mathbf{V}_0$, where V_0 is the nominal velocity command.
- This can be accomplished as follows:
  - Compute a random noise vector **n** from a uniform distribution
  - Compute $\mathbf{r}_{noise} = K_{noise} \dfrac{\mathbf{n}}{\|\mathbf{n}\|}$ where K_{noise} is the noise scale factor
  - Compute the new wander velocity given the current wander velocity, $\mathbf{V}_{wander}$, and the wander direction perturbation $\mathbf{r}_{noise}$,
    - $\mathbf{V}_{wander} = K_{wander} \dfrac{\mathbf{V}_{wander} + \mathbf{r}_{noise}}{\|\mathbf{V}_{wander} + \mathbf{r}_{noise}\|}$ where K_{wander} is the wander strength
  - Compute the desired velocity vector as
    - $\mathbf{V}_{desired} = \mathbf{V}_0 + \mathbf{V}_{wander}$
  - Knowing $\mathbf{V}_{desired}$, compute command variables V_d and θ_d as in Seek behavior