KUTAISI
INTERNATIONAL
UNIVERSITY

Numerical Programming

Ramaz Botchorishvili

Kutaisi International University

Final, Retake Project - Simulation of autonomous navigation

Problem statement

Problem 0.1

Simulation of autonomous navigation using numerical modeling of a swarm motion in a constrained environment. Three sub-problems must be solved:

1. Follow a path in a constrained environment, 5 points.
   - Input: Open a map, choose two points A and B connected with a path, take a snapshot. Assume map area is flat, assign a width to a path, assign a size to your robot.
   - Task: extract path from A to B and parametrize using splines. A robot must move from A to B following the path and without crossing borders of the path.
   - Output: path with width and visualization of robot following the path.
2. Swarm of robots in a constrained path, 5 points
   - Input: A path and its width, swarm of robots at A and at B.
   - Task: swarms of robots must autonomously navigate from A to B and from B to A without accidents. Motion from A and B must start at the same time.
   - Output: path with width and visualization of robots navigation without accident.
3. Navigation in a pedestrian flow, 5 points
   - Input: A pedestrian flow video of your choice.
   - Task: a robot must safely navigate in pedetsrian flow in at least two directins.
   - Output: visualization of navigation including robot and pedestrians.

Notes on project implementation

Important remarks

To guarantee a successful implementation of this computational project, the student must make a series of interdependent decisions concerning the mathematical model, numerical methods, input data (including pre-processing), validation and verification, as well as output interpretation and communication.

- The student must develop or select a mathematical model that describes collision-free motion of a swarm. Several candidate models are presented on the slides below; choosing and refining the appropriate model is the student's responsibility.
- The student must determine the number of robots required for each sub-problem. whereas a two-dimensional motion is governed by four equations instead of six in three dimensions.
- Any built-in library may be employed for visualization.
- Student must be familiar and comprehend essential methods and approaces used in the project implementation.

General requirements, common tasks for sub-problems

Tasks

- For each sub-problem formulate suitable BVP or IVP.
- Clearly describe all inputs to the selected equations including initial conditions, boundary conditions, and user defined parameters.
- Formulate numerical methods, algorithms, and their properties, explain and justify your approach in written.
- Develop test cases and demonstrate validity of your results.
- Upload all necessary files, including
  1. Presentation file
  2. Code
  3. Test data and their description
  4. Visualisation
  5. Test cases for which your approach does work well, and does not work, with explanations of limitations.

Initial Value Problem (IVP)

IVP

- The state of robot i is governed by its position $\mathbf{x}_i \in \mathbb{R}^3$ and velocity $\mathbf{v}_i \in \mathbb{R}^3$, $i = 1, \ldots, N$
- Governing equations with velocity saturation:

$$\dot{\mathbf{x}}_i = \mathbf{v}_i \cdot \min\left(1, \frac{v_{max}}{\|\mathbf{v}_i\|}\right)$$

$$\dot{\mathbf{v}}_i(t) = \frac{1}{m}\left[k_p\left(\mathbf{T}_i(t) - \mathbf{x}_i(t)\right) + \sum_{j \neq i} \mathbf{f}_{\text{rep}}(\mathbf{x}_i(t), \mathbf{x}_j(t)) - k_d \mathbf{v}_i(t)\right],$$

- Initial conditions:

$$\mathbf{x}_i(0) = \mathbf{x}_{i,0}, \quad \mathbf{v}_i(0) = \mathbf{v}_{i,0},$$

  - **Target Tracking:** $\mathbf{T}_i(t)$ represents the time-varying waypoint for a robot i.
  - **Collision Avoidance:** $\mathbf{f}_{rep}$ is a repulsive force (e.g., inverse-square law) that activates when robots are within a safety radius R_{safe}.
  - **Damping:** k_d ensures the system reaches a steady state without infinite oscillation.
  - More on notations with explanation is given below.

Boundary Value Problem (BVP)

BVP

- The state of robot i is governed by its position $\mathbf{x}_i \in \mathbb{R}^3$ and velocity $\mathbf{v}_i \in \mathbb{R}^3$, $i = 1, \ldots, N$
- Governing equations with velocity saturation:

$$\dot{\mathbf{x}}_i = \mathbf{v}_i \cdot \min\left(1, \frac{v_{max}}{\|\mathbf{v}_i\|}\right)$$

$$\dot{\mathbf{v}}_i(t) = \frac{1}{m}\left[k_p\left(\mathbf{T}_i(t) - \mathbf{x}_i(t)\right) + \sum_{j\neq i} \mathbf{f}_{\mathrm{rep}}(\mathbf{x}_i(t), \mathbf{x}_j(t)) - k_d \mathbf{v}_i(t)\right],$$

- Boundary conditions:

$$\mathbf{x}_i(0) = \mathbf{x}_{i,0}, \quad \mathbf{x}_i(T) = \mathbf{x}_{i,T},$$

- **Target Tracking:** $\mathbf{T}_i(t)$ represents the time-varying waypoint for robot i corresponding to the desired track. You can consider several sub-models where T_i is time independent, $k_d = 0$ and (or) $k_p = 0$.
- **Collision Avoidance:** $\mathbf{f}_{rep}$ is a repulsive force (e.g., inverse-square law) that activates when robots are within a safety radius R_{safe}.
- More on notations with explanation is given below.

Initial Value Problem with Velocity Tracking (IVP VT)

IVP VT

- The state of robot i is governed by its position $\mathbf{x}_i \in \mathbb{R}^3$ and velocity $\mathbf{v}_i \in \mathbb{R}^3$, $i = 1, \ldots, N$.
- $$\mathbf{V}_{\text{sat}}(\mathbf{x}, t) = \mathbf{V}(\mathbf{x}, t) \cdot \min\left(1, \frac{v_{\max}}{\|\mathbf{V}(\mathbf{x}, t)\|}\right) \quad \text{if } \|\mathbf{V}(\mathbf{x}, t)\| > 0, \quad \mathbf{0} \quad \text{otherwise.}$$
- Governing equations with velocity saturation:

$$\dot{\mathbf{x}}_i = \mathbf{v}_i \cdot \min\left(1, \frac{v_{max}}{\|\mathbf{v}_i\|}\right)$$

$$\dot{\mathbf{v}}_i(t) = \frac{1}{m}\left[k_v\left(\mathbf{V}_{\text{sat}}(\mathbf{x}_i(t), t) - \mathbf{v}_i(t)\right) + \sum_{j \neq i} \mathbf{f}_{\text{rep}}(\mathbf{x}_i(t), \mathbf{x}_j(t)) - k_d \mathbf{v}_i(t)\right],$$

- Initial conditions:

$$\mathbf{x}_i(0) = \mathbf{x}_{i,0}, \quad \mathbf{v}_i(0) = \mathbf{v}_{i,0},$$

Parameters and variables

- saturated velocity field:

$$\mathbf{V}_{\text{sat}}(\mathbf{x}, t) = \mathbf{V}(\mathbf{x}, t) \cdot \min\left(1, \frac{v_{\max}}{\|\mathbf{V}(\mathbf{x}, t)\|}\right) \quad \text{if } \|\mathbf{V}(\mathbf{x}, t)\| > 0, \quad \mathbf{0} \quad \text{otherwise.} \tag{1}$$

- $\mathbf{x}_i \in \mathbb{R}^d$: Position of robot i.
- $\mathbf{v}_i \in \mathbb{R}^d$: Velocity of robot i.
- $m > 0$: Mass of each robot (constant, assumed uniform).
- $\mathbf{V}(\mathbf{x}, t) \in \mathbb{R}^d$: Velocity field extracted from the reference video.
- $\mathbf{V}_{\text{sat}}(\mathbf{x}, t) \in \mathbb{R}^d$: Saturated velocity field with $\|\mathbf{V}_{\text{sat}}\| \leq v_{\max}$.
- $v_{\max} > 0$: Maximum allowable velocity magnitude (scalar, hardware-dependent).
- $k_v > 0$: Velocity-matching gain.
- $k_d > 0$: Damping coefficient.
- $k_{\text{rep}} > 0$: Repulsion gain.
- $R_{\text{safe}} > 0$: Safety radius.
- $\mathbf{f}_{\text{rep}}(\mathbf{x}_i, \mathbf{x}_j) \in \mathbb{R}^d$: Pairwise repulsive force.

$$\mathbf{f}_{\text{rep}}(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} k_{\text{rep}} \dfrac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|^3} & \text{if } \|\mathbf{x}_i - \mathbf{x}_j\| < R_{\text{safe}}, \\ \mathbf{0} & \text{otherwise.} \end{cases} \tag{2}$$

- N: Number of robots.
- d: Spatial dimension (typically 2).

Final Remarks

Important Notice

- For successfully solving the problem student must know the following: derivatives and their approximations, edge detection, splines, IVP for ODEs, BVP for ODEs, RK methods and Butcher's table, truncation error, A-stability, shooting methods, numerical solution of linear and nonlinear systems of equations, optical flow.
- Student must explicitly mention using of AI and provide all related necessary details.
- The project is assigned 0 points if:
  - Any of the requested and/or necessary file is missing.
  - Submitted results are not reproducible.
  - Student cannot apply his own code for the input data provided by TA or instructor.
- Submission deadline: will be aligned with the schedule of final exams. See corresponding assignment.