Loop가 존재하는 미로 탐색 방안

유레카 7조(A-Maze-ing)

Kookmin University

November 13, 2025

Table of Contents

1. Introduction
2. Definitions
3. Trémaux's algorithm
4. Guarantees for the Trémaux's algorithm
5. Sliding Mode Control
6. Reference

Introduction

질문: 누군가 물에 들어오라고 묻는다.

- 선택1: 일단 따라서 들어간다
- 선택2: 어떤 물인지 확인한다

Introduction – 문제 파악

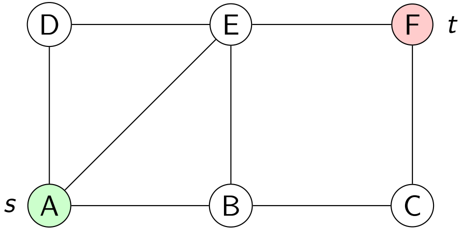
- 맨 몸으로 태평양에 끌려가면 죽을 수 있다
- 적절한 대응 방안을 미리 생각하는 것이 중요
- 미로 탐색도 그렇다.

Definition of a Maze

Maze (Graph Representation)

A maze is an undirected graph $G = (V, E)$ where:

- V: finite set of nodes (cells)
- $E \subseteq \{\{u, v\} \mid u, v \in V,\ u \neq v\}$: set of edges (passages)
- $s \in V$: starting node
- $t \in V$: goal (exit) node



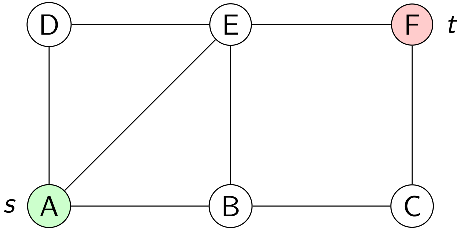
Definition of Path

Exploration Path

A path from s to t is:

$$P = (v_0, v_1, \ldots, v_k)$$

where $v_0 = s,\ v_k = t,\ (v_i, v_{i+1}) \in E\ \forall\ 0 \leq i < k.$



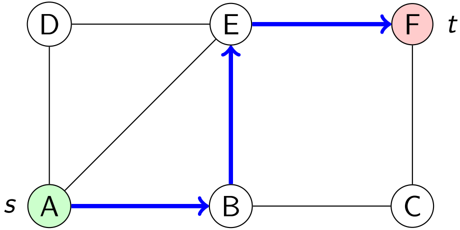
Example of Exploration Path

Exploration Path

A path from s to t is:

$$P = (A, B, E, F)$$

where $A = s$, $F = t$, and each consecutive pair is connected by an edge in E.



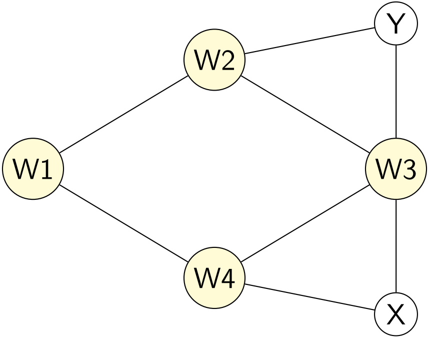
Definition of Loop(cycle)

Loop (Cycle)

A cycle [Gallier and Quaintance, 2024] is a sequence:

$$(w_0, w_1, \ldots, w_m, w_0), \quad m \geq 2$$

with distinct nodes (except start/end), where $(w_i, w_{i+1}) \in E \; \forall \; 0 \leq i < m$.



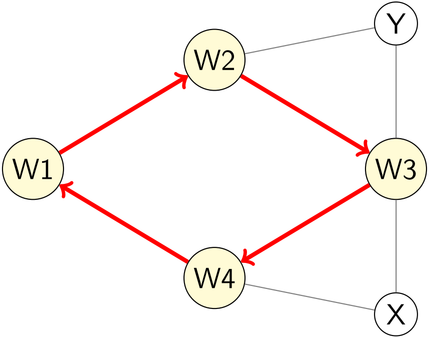
Example of Loop (Cycle)

Cycle in a Maze Graph

A cycle is a sequence:

$$(w_1, w_2, w_3, w_4, w_1)$$

where each node is distinct (except start/end), and each consecutive pair is connected by an edge.



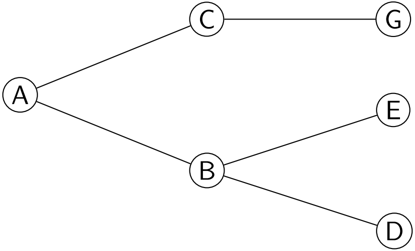
Classical Left-Hand Rule

Guarantees for the Left-Hand Rule

- If the maze boundary is a single continuous curve (no cycles), the rule guarantees reaching the exit.
- Formally: If G is a tree (connected and acyclic),

$$G \text{ is a tree} \implies \text{left-hand rule always leads to the exit}$$

- Topologically: $\pi_1(G) = 0$ (trivial fundamental group).



Mazes with Loops (Cycles)

Why the Left-Hand Rule Can Fail

- If G contains cycles, the left-hand rule may cause endless tracing of a cycle boundary.
- Formally:

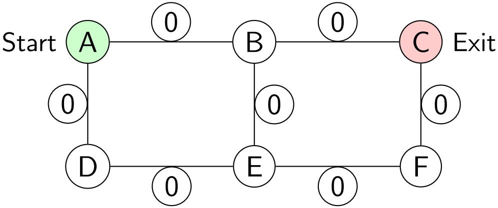
$$\exists \text{ cycle } C \subseteq G \text{ such that the left-hand rule can follow } C \text{ indefinitely}$$

- The explorer may fail to reach the exit depending on the starting position and maze structure.

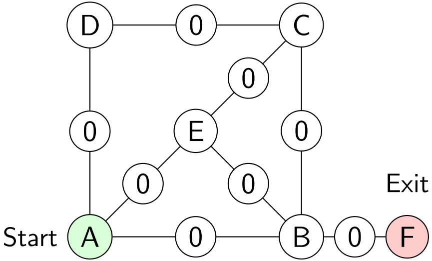
Definition of Trémaux Edge

Trémaux Edge Marking

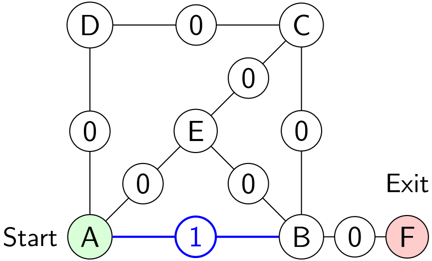
Each edge tracking: $f : E \to \{0, 1, 2\}$ counts visits (up to twice per edge).



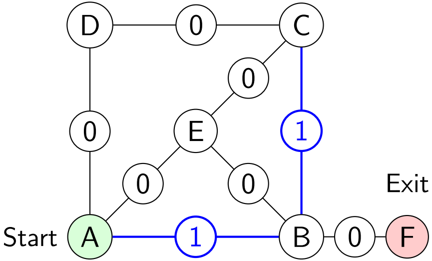
Tremaux Edge Marking: Step 1



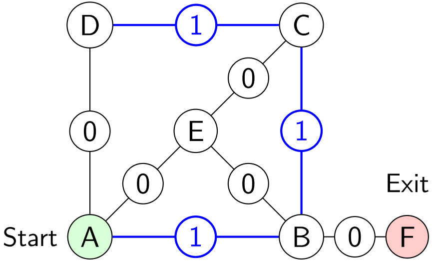
Tremaux Edge Marking: Step 2



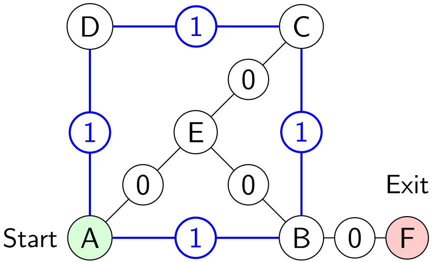
Tremaux Edge Marking: Step 3



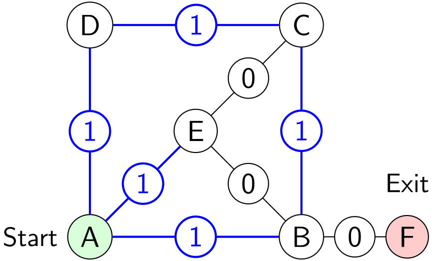
Tremaux Edge Marking: Step 4



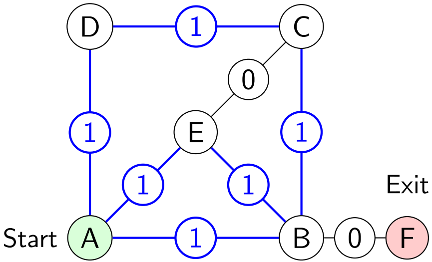
Tremaux Edge Marking: Step 5



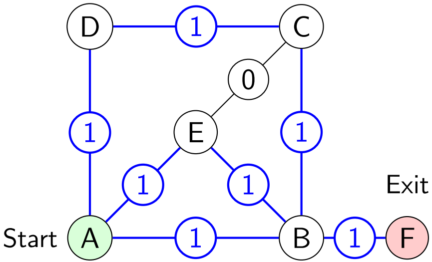
Tremaux Edge Marking: Step 6



Tremaux Edge Marking: Step 7



Tremaux Edge Marking: Step 8



Trémaux's algorithm

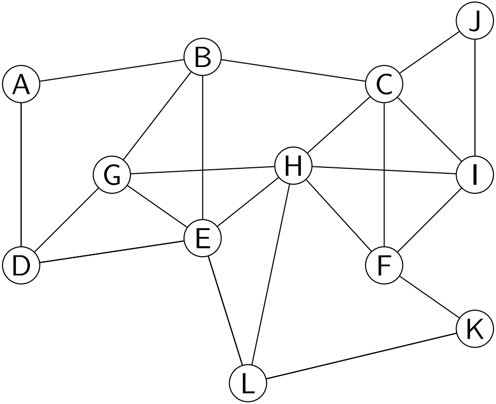
while *not at goal* **do**
- Detect intersection i;
- **if** $V[i] = 0$ **then**
  - Choose an unvisited path;
  - Move forward;
  - $V[i] \leftarrow 1$;
- **end**
- **else if** $V[i] = 1$ **then**
  - Choose a different path than before;
  - Move forward;
  - $V[i] \leftarrow 2$;
- **end**
- **else**
  - Backtrack to previous intersection;
- **end**

end

Algorithm 1: Trémaux Maze Solving Algorithm

의문

- 정말 loop가 있는 모든 미로를 빠져나갈 수 있는가?

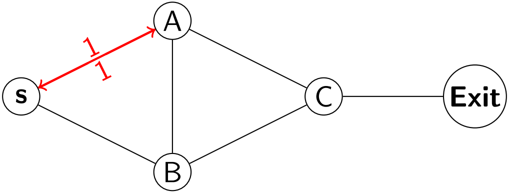


- 하나, 둘, 셋, ... 많다

Traversal Bound Lemma

Lemma (Traversal Bound)

For all $e \in E$, $M(e) \leq 2$; *thus* $\sum_{e \in E} M(e) \leq 2|E|$.



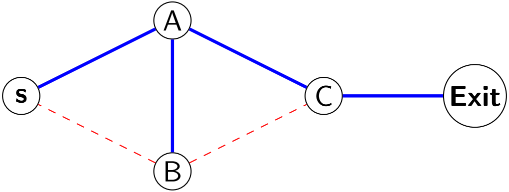
Proof.

Each edge direction (u, v) may only be marked once, by the rules: advance and (optionally) backtrack. After both directions are marked ($M(e) = 2$), the edge is never traversed again. Thus, no edge exceeds two total traversals. □

DFS Structure Lemma

Lemma (DFS Spanning Tree)

The forward traversals induce a DFS spanning tree T of G rooted at s; non-tree edges are back edges joining ancestor-descendant pairs.



Proof.

The advance rule always follows unmarked edges, producing a tree. Backtracks occur only when no advances are possible, exactly mimicking the DFS algorithm. Non-forward traversed edges connect previously visited vertices (ancestor-descendant relations). □

Main Theorem: Termination

Theorem (Termination)

The Trémaux algorithm terminates in at most $2|E|$ *steps.*

Proof.

Each step marks $m(u, v)$ from 0 to 1 for some directed edge. There are $2|E|$ possible directed markings in G. When all are marked, the algorithm halts. Thus, execution is always finite. □

- 어떤 복잡한 미로라도 무한 탐색을 하지 않는다!

Reachability and Loop Avoidance

Theorem (Loop Avoidance)

No edge is traversed more than twice; the algorithm cannot become trapped in any cycle.

Proof.

After a cycle's entry and return, all edges in the cycle reach $M(e) = 2$ and are no longer available for traversal. Thus, no infinite looping within cycles can occur. □

- loop에 빠지지 않는다!

Reachability and Loop Avoidance

Theorem (Reachability)

If $X \cap C(s) \neq \emptyset$ ($C(s)$ is the connected component containing s), some $x^* \in X \cap C(s)$ is visited in finite time.

Proof.

By the DFS equivalence lemma, the algorithm traverses the entire component $C(s)$ via the induced spanning tree T. Every vertex in $C(s)$ and, in particular, every exit $x^* \in X \cap C(s)$ is eventually reached. $\square$

- 출구가 있다면 도달한다!

Exit Path Recovery

Theorem (Unique Exit Path)

Upon first reaching an exit x^*, the path from s to x^* in the induced DFS tree T is unique and recoverable using the markings m.

Proof.

The unique tree path $\pi = (v_0 = s, v_1, ..., v_k = x^*)$ corresponds to a sequence of edges with $m(v_i, v_{i+1}) = 1$ (forward marks). As the DFS tree is unique (from Trémaux's rules), the solution path is uniquely encoded in m. □

- 입구–출구 경로를 알 수 있다

Why control needed?

- 미로 내 안정적인 이동이 필요하다
- PID: 외란에 취약하며, linear control
- SMC: 외란에 강인하며, nonlinear control

System Model

Consider a general second-order nonlinear system:

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= f(x) + g(x)u + d(t)\end{aligned}$$

where x_1 is the state, u is the control input, and $d(t)$ denotes disturbances.

Sliding Surface Definition

The sliding surface is constructed as:

$$s = \left(\frac{d}{dt} + \lambda\right)^{n-1} e(t)$$

If $n = 2$ (second-order):

$$s = \dot{e}(t) + \lambda e(t)$$

where $e(t) = x_1(t) - x_1^{\mathrm{d}}$ is the tracking error, and $\lambda > 0$ is a design parameter.

Control Law

A typical SMC law is:

$$u = u_{eq} - K \cdot \mathrm{sign}(s)$$

For chattering reduction, replace with a smooth function:

$$u = u_{eq} - K \cdot \tanh\left(\frac{s}{\varepsilon}\right)$$

where $K > 0$ and $\varepsilon > 0$.

Error Dynamics

Consider a second-order error system:

$$e_1 = x_1 - x_1^{\mathrm{d}}$$
$$e_2 = x_2 - x_2^{\mathrm{d}}$$

Define the sliding surface as:

$$s = e_2 + \lambda e_1$$

where $\lambda > 0$.
Error dynamics:

$$\dot{e}_1 = e_2$$
$$\dot{e}_2 = f(x) + g(x)u + d(t) - x_2^{\mathrm{d}}$$

Finite Reaching Time

On the sliding surface ($s \to 0$):

$$e_2 = -\lambda e_1 \implies \dot{e}_1 + \lambda e_1 = 0$$

Finite-time reachability:

$$V = \frac{1}{2}s^2, \qquad \dot{V} = s \cdot \dot{s} \leq -\eta|s|, \qquad \eta > 0$$

Therefore,

$$t_{\text{reach}} \leq \frac{|s(0)|}{\eta}$$

After t_{reach}, system will remain on $s = 0$ if unmatched disturbances are bounded.

Stability and Reaching Condition

Lyapunov function:

$$V = \frac{1}{2}s^2$$

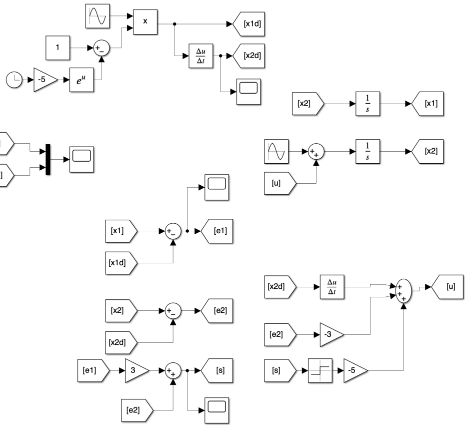
To guarantee reachability and stability, require:

$$\dot{V} = s\dot{s} \le -\eta|s|, \qquad \eta > 0$$

The sliding mode exists if the control gain K is large enough:

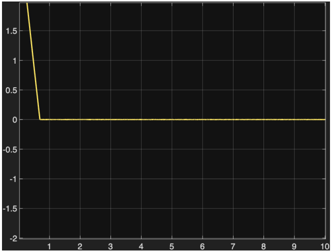
$$K > \max\left|f(x) + d(t)\right|$$

Testing SMC – Controller design

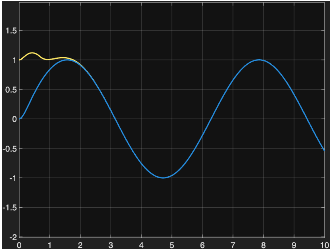


- 간단한 2차 시스템 가정
- 추종 입력 sine 함수, 외란 존재 가정

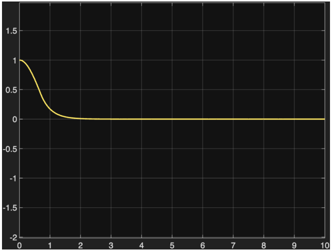
Testing SMC – Controller simulation



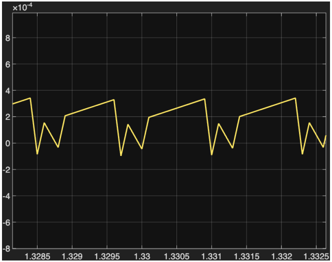
Error



Sliding



Sliding?



Chattering

Summary

- Trémaux 알고리즘은 출구를 찾게 해준다
- SMC은 외란에 강인하다
- Design parameter λ와 control gain K를 잘 조절해야 한다
- 강인성을 높이려면 채터링을 감수해야한다(trade-off)

github

https://github.com/Woosang-Cho/amazing

Reference I

Gallier, J. H. and Quaintance, J. (2024).
Mathematical Foundations and Aspects of Discrete Mathematics.
University of Pennsylvania, Philadelphia.
Available as a PDF ebook.

Thanks!

physiklove@kookmin.ac.kr