

令和 7 年度 卒業論文

# マニピュレータのなぞり動作とミリ波レーダ センサを用いた構造物内部の可視化

2025 年 11 月 25 日

宇都宮大学  
基盤工学科  
機械システム工学コース

学籍番号：222268A  
氏名：小林 暖弥

指導教員  
尾崎 功一 教授  
ミヤグスク レナート 准教授  
田畠 研太 助教

# 目次

<b>第 1 章 緒言</b>	<b>1</b>
1.1 研究の背景	1
1.2 本研究の目的	1
1.3 本論文の構成	2
<b>第 2 章 関連研究</b>	<b>3</b>
2.1 物体をなぞる操作の必要性	3
2.2 物体をなぞる操作に関する従来研究	3
2.3 まとめ	4
<b>第 3 章 なぞり動作とレーダセンサによる構造物内部の可視化</b>	<b>5</b>
3.1 エンドエフェクタの設計	5
3.2 走査するレーダセンサの選定	5
3.3 なぞり動作のアルゴリズム	6
3.4 可視化システム構成	7
<b>第 4 章 ソフトウェア設計</b>	<b>9</b>
4.1 カルマンフィルタの実装	9
4.2 ソフトウェア評価	10
<b>第 5 章 実環境におけるシステム評価</b>	<b>11</b>
5.1 実験目的	11
5.2 実験環境	11
5.3 実験結果	11
5.3.1 現在の位置姿勢	11
5.3.2 次の位置姿勢	11
5.3.3 更新	12
5.4 考察	12
<b>第 6 章 シミュレーション評価</b>	<b>13</b>
6.1 シミュレーション環境の構築	13
6.2 実験概要	13
6.3 考察	13
<b>第 7 章 結言</b>	<b>15</b>

---

7.1 結論	15
7.2 課題	15
7.3 展望	15
<b>謝辞</b>	<b>16</b>
<b>付録 A 詳細な電子回路図</b>	<b>18</b>
<b>付録 B 主要なプログラムのソースコード</b>	<b>19</b>
B.1 C++	19
B.2 Python	19
<b>付録 C 追加実験</b>	<b>21</b>
C.1 実験 A	21
C.2 実験 B	21

## 図目次

3.1 Appearance of the end effector	5
3.2 Endeffector coordinate system	7
3.3 Calculation symbols	8
3.3(a) Distribution of sensor data	8
3.3(b) Center of mass, velocity, and angular velocity	8
A.1 システム全体の回路図	18

## 表目次

6.1 性能比較	14
----------	----

# 第 1 章

## 緒言

### 1.1 研究の背景

日本のインフラは高度成長期以降に集中的に整備されたことから、今後 20 年間で建設後 50 年以上経過する施設の割合が加速度的に高くなると見込まれており、一斉に老朽化する社会資本を維持管理していくことが課題となっている [1]. 一方で、建設業就業者はピーク時から 200 万人減少、就業者の 3 割以上が 55 歳以上と高齢化も進み、労働力不足が深刻化している [2]. こうしたことから、インフラメンテナンスの自動化、省人化が進められている。特に非破壊検査の領域では、従来の手作業に比べて高速かつ高精度な点検が可能となることや、データの収集と解析を効率的に行えるようになることから、ロボットの導入が進められている。

非破壊検査を行うロボットの主な形態として、無人航空機 (UAV), 地上無人車両 (UGV) が挙げられる。UAV は、機動性が高く人間がアクセスしにくい場所でもデータを収集できるが、風などの外部環境に左右されやすいほか、ペイロードが小さく搭載できる機器や活動時間が制限されるため、一度のフライトでは多様な検査が行えない。また、UGV はペイロードが大きく様々なセンサを搭載できるが、検査の対象は一般的に接地面に限られる [3]. そこで、こうした弱点を補完し、一台で様々な検査が可能となるロボットの形態として、モバイルマニピュレータが注目されている。

モバイルマニピュレータは、ロボットマニピュレータを UGV などのモバイルプラットフォームに搭載したシステムのことである [4]. これを非破壊検査に応用すると、マニピュレータが壁面を、UGV が接地面を検査することによって、精密で広域な検査が可能となる。また、マニピュレータのエンドエフェクタを付け替えることにより多様なセンサによるマルチな検査を行うことができる。

### 1.2 本研究の目的

モバイルマニピュレータが検査を行う際、マニピュレータは UGV の動作に合わせて適切に検査対象面をなぞりながらセンサを走査させる必要があるため、その制御が非常に重要である。本研究では、そのなぞり動作を近接覚センサを用いたマニピュレーションによって実

現する。加えて、水道管や空洞、埋没物の非破壊検査を念頭に置き、ミリ波レーダセンサを走査して構造物内部を可視化するシステムを実現する。

### 1.3 本論文の構成

本論文の構成を以下に述べる。

本章では、研究背景および研究目的について述べた。

第 2 章では、物体なぞる操作の必要性と、なぞり動作の関連研究について述べる。

第 3 章では、  
きききききききききききききききききききききききききききききき  
きききききききききききききききききききききききききききききき  
ききききききききききききききききききききききききききききき  
ききききききききききききききききききききききききききききき  
ききききききききききききききききききききききききききききき.

第 4 章では、  
くくくくくくくくくくくくくくくくくくくくくくくくくくくくくく  
くくくくくくくくくくくくくくくくくくくくくくくくくくくくくく  
くくくくくくくくくくくくくくくくくくくくくくくくくくくくく.

第 5 章では、  
けけけけけけけけけけけけけけけけけけけけけけけけけけけけ  
けけけけけけけけけけけけけけけけけけけけけけけけけけけけ  
けけけけけけけけけけけけけけけけけけけけけけけけけけけけ  
けけけけけけけけけけけけけけけけけけけけけけけけけけけけ.

第 6 章では、  
ここここここここここここここここここここここここここ  
ここここここここここここここここここここここここここ  
ここここここここここここここここここここここここ  
ここここここここここここここここここここここ.

第 7 章では、  
わわわわわわわわわわわわわわわわわわわわわわわわわ  
わわわわわわわわわわわわわわわわわわわわわわわわわ  
わわわわわわわわわわわわわわわわわわわわわわわ  
わわわわわわわわわわわわわわわわわわわわわわわわ.

# 第 2 章

## 関連研究

### 2.1 物体をなぞる操作の必要性

インフラ内部の欠陥を可視化する手法として、地中レーダ (Ground Penetrating Radar : GPR) による検査が広く用いられている。GPR は、装置を移動させながらおよそ 50MHz~4.5GHz の電磁波を物体中に伝搬させ、内部構造物からの反射波の周波数毎の時間、強度、波形を計測することでその内部を測定する [5]。一般に、GPR を含むレーダセンサは、物体表面から遠すぎると物体中への電波の放射が弱まり、探査深度が大幅に減少するなど測定の精度が低下する [6]。そのため、検査を行う際にはレーダセンサを測定対象の表面に十分近接させつつ接触しないように走査させる必要がある。

### 2.2 物体をなぞる操作に関する従来研究

Chen らは、複雑な形状の物体の塗装のため、点群データを用いてマニピュレータの軌道計画を行った [7]。この研究では、点群モデルを取得し、これをスライスして物体の輪郭に対する法線ベクトルを推定することで、物体をなぞるマニピュレーションを実現した。しかし、この手法では軌道計画のために点群データの取得と処理を事前に行う必要があり、リアルタイム性が重要となるモバイルマニピュレータには向かない。検査のためには、自己位置推定やナビゲーションの誤差によって UGV と壁面との距離が変化しても、リアクティブにその誤差を吸収する必要がある。

リアルタイム性を持つ操作として、水内らは近接覚センサ情報に基づいてキッチン組み込み型ロボットが“なぞり動作”を行い、未知の食器形状を把握することを可能にした [8]。この研究では、ロボットハンド表面の同一平面上に赤外線の発光・受光素子を一体化した近接覚センサ 5 点を埋め込み、これらの信号強度が均一かつ一定に保つことで次の 3 つの動作を行った。

- (1) 食器表面とセンサの向きを一致させる動作
- (2) 食器表面との間隔を調整する動作
- (3) センサ面と平行に移動する動作

これらによって実現される“なぞり動作”を応用することで、レーダセンサを物体表面に対

して十分に近接させながら走査させるマニピュレーションが達成できると考えられる。

## 2.3 まとめ

本章では、まず GPR を含むレーダセンサの特性について述べ、それらを測定対象の表面に十分近接させつつ接触しないように走査させるため、物体をなぞる操作が必要であることを示した。また、なぞり動作に関する従来研究を紹介し、リアルタイム性を持ったマニピュレーションが求められることから、複数の赤外線近接覚センサを用いた“なぞり動作”が検査に有効であると示した。これを踏まえ、本研究では、なぞり動作によってレーダセンサを走査し、構造物内部の物体を可視化するシステムを実装した。次章では、その詳細について述べる。

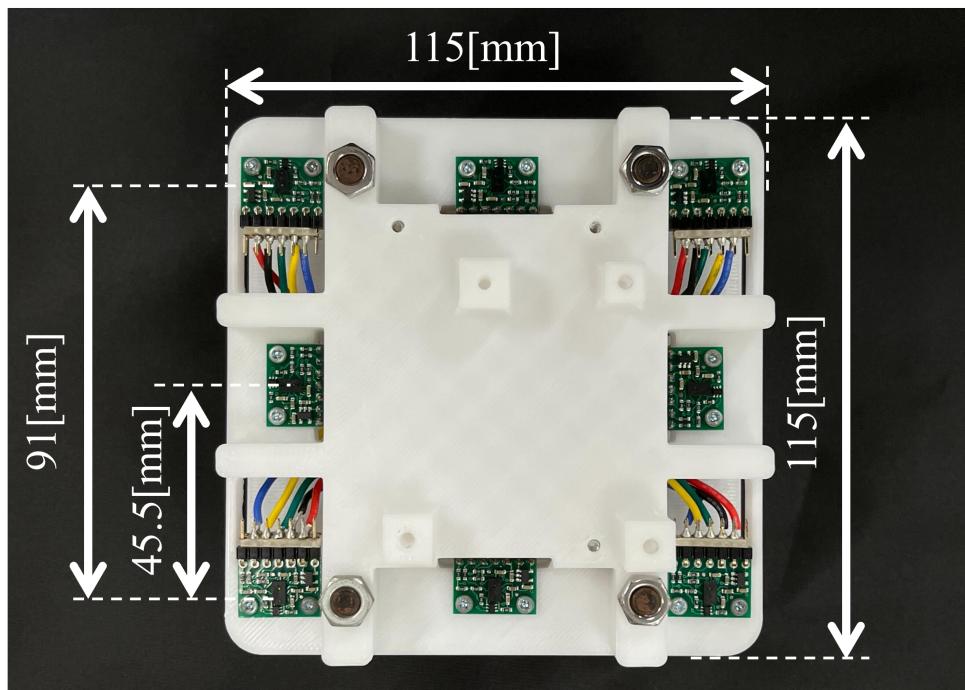


Fig. 3.1 Appearance of the end effector  
**第3章**

## なぞり動作とレーダセンサによる構造物内部の可視化

### 3.1 エンドエフェクタの設計

製作したエンドエフェクタの外観を 3.1 に示す。エンドエフェクタには、赤外線近接距離センサモジュール (pololu, VL6180X) が同一平面上の  $91\text{mm} \times 91\text{mm}$  の正方形の頂点に 4 つと、各辺の中心に 4 つの計 8 つ設置されている。この配置は、エンドエフェクタの姿勢の変化を安定して取得しつつ、後述するセンサ情報の計算を容易にするために設計されている。なお、これらの近接覚センサはその位置の重心がエンドエフェクタの中心と一致するよう取り付けられている。また、中央部にはレーダセンサを取り付けるための支柱と穴が設けられている。近接覚センサは  $0 \sim 255\text{mm}$  の範囲で  $1\text{mm}$  の分解能で距離を測定するが、各モジュールごとに得られる距離値に個体差があるため、使用する際はオフセットを設定してそれぞれの値を均一化する。

## 3.2 走査するレーダセンサの選定

レーダセンサの解像度はレーダの波長に依存しているため、GPR では 5cm 未満の物体が「点」として検出されたり、数 cm 間隔で並ぶ鉄筋が別個のオブジェクトとして認識されないなどといった問題があった [9][10]. 一方で、波長が短く周波数が高いレーダセンサは物体中で減衰しやすく探査深度が浅くなってしまうことからインフラ等の検査にはあまり使用されない. しかし、本研究ではより高解像度に物体中の構造を可視化することを目指してミリ波レーダを用いる. 中でも、活用の際の法律的なハードルが低いことから 60GHz 帯のレーダを使用する. ミリ波レーダは、acconeer 社製の A121 レーダセンサチップを搭載した評価キット XE121 と、コネクタボード XC120 を用いる.

## 3.3 なぞり動作のアルゴリズム

なぞり動作は、エンドエフェクタを取り付けた 7 自由度協働ロボットアーム (UFactory, xArm7) によって行う. エンドエフェクタには 3.2 に示すように座標系が定義されており、これにしたがって近接覚センサの情報を計算して速度および角速度を与える. なお、速度と角速度は近接覚センサの誤差による振動を抑えるために PD 制御されている.

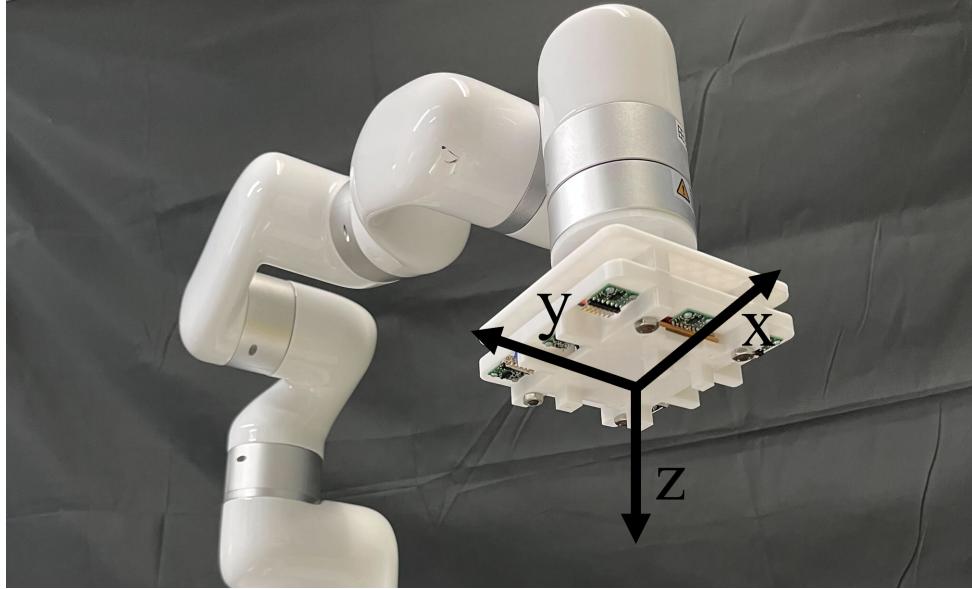
以上の計算に用いるパラメータを、3.3 に示す. エンドエフェクタの面上に  $N$  個の近接覚センサが分布しており、その面の法線ベクトル  $\mathbf{n}$  の方向を計測しているとき、エンドエフェクタ座標系における各センサの位置を  $\mathbf{p}_i$ 、各センサから得られる計測距離を  $s_i$  とすると ( $i = 1, 2, \dots, N$ )、センサの位置の重心  $\mathbf{p}_C$  およびセンサの計測距離の重心  $\mathbf{p}_{SG}$  は次式で表される.

$$\mathbf{p}_C = \frac{\sum_{i=1}^N \mathbf{p}_i}{N}, \quad \mathbf{p}_{SG} = \frac{\sum_{i=1}^N s_i \mathbf{p}_i}{\sum_{i=1}^N s_i} \quad (3.1)$$

センサ情報  $s_i$  に基づき、このセンサ位置の重心  $\mathbf{p}_C$  に対して物体表面のなぞり動作を行うための目標速度  $v$  と目標角速度  $\omega$  を与える. 目標速度  $v$  は次式となる.

$$\mathbf{v} = \mathbf{v}_{trace} + \mathbf{v}_{ctrl} \quad (3.2)$$

ここで、 $\mathbf{v}_{trace}$  は物体に沿ってセンサ面に平行な方向にエンドエフェクタを動かすための速度の指令値である. また、 $\mathbf{v}_{ctrl}$  は物体表面とセンサ面の距離を一定に保つための速度であり、PD 制御されている. これらは、近接覚センサが計測した距離の最小値  $s_{min}$  と、センサ面と物体表面の目標距離  $s_{target}$  の差  $s_e$  に基づいて次のように決定される. なお、 $k_{vP}$  と  $k_{vD}$



**Fig. 3.2** Endeffector coordinate system

はそれぞれ比例ゲインと微分ゲインである。

$$s_{min} = \max_i s_i \quad (3.3)$$

$$s_e = s_{min} - s_{target} \quad (3.4)$$

$$\mathbf{v}_{ctrl} = k_{vP}s_e\mathbf{n} + k_{vD}(s_e - s_{e-1})\mathbf{n} \quad (3.5)$$

次に、物体表面とセンサ面を平行にするための回転速度である目標角速度  $\omega$  は次式で表される。なお、 $k_{\omega P}$  と  $k_{\omega D}$  は各比例ゲインと微分ゲインである。

$$\mathbf{p}_e = \mathbf{p}_C - \mathbf{p}_{SG} \quad (3.6)$$

$$\boldsymbol{\omega} = k_{\omega P}\mathbf{n} \times \mathbf{p}_e + k_{\omega D}\mathbf{n} \times (\mathbf{p}_e - \mathbf{p}_{e-1}) \quad (3.7)$$

$\mathbf{p}_C - \mathbf{p}_{SG}$  は計測距離の重心  $\mathbf{p}_{SG}$  から位置の重心  $\mathbf{p}_C$  に向かうセンサ面内のベクトルであるから、位置の重心  $\mathbf{p}_C$  を中心としてエンドエフェクタに  $\mathbf{n} \times (\mathbf{p}_C - \mathbf{p}_{SG})$  方向の回転をさせると  $\mathbf{p}_{SG}$  が物体表面に近づくように動き、その結果  $\mathbf{p}_{SG}$  と  $\mathbf{p}_C$  が一致する動きをとることが期待される。これにより、なぞり動作が実現される。

## 3.4 可視化システム構成

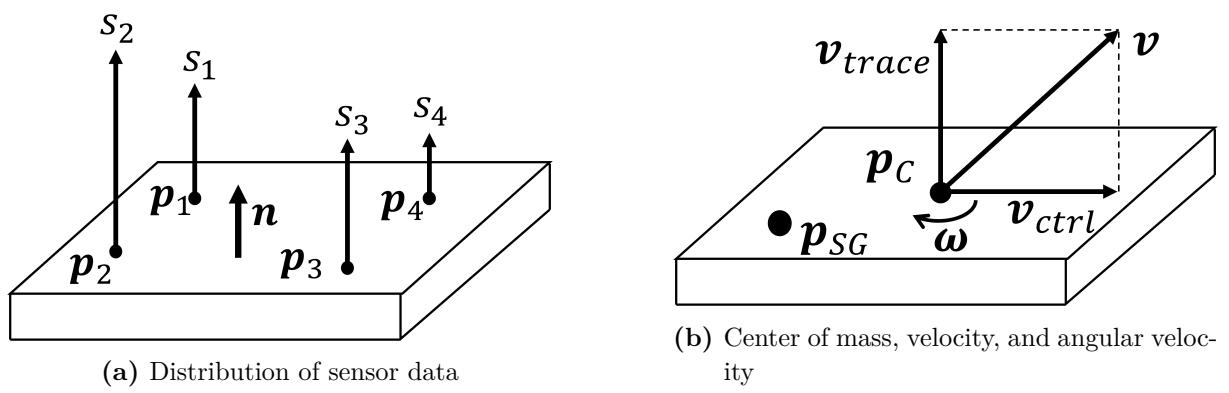


Fig. 3.3 Calculation symbols

# 第4章

## ソフトウェア設計

## 4.1 カルマンフィルタの実装

---

**Algorithm 1** Kalman Filter

---

- ```

1: Input: Initial State  $\hat{\mathbf{x}}_{0|0}$ , Initial Covariance  $\mathbf{P}_{0|0}$ 
2: Output: Estimated State at time  $k$ ,  $\hat{\mathbf{x}}_{k|k}$ 
3:
4: for  $k = 1, 2, \dots$  do
5:   Prediction Step
6:      $\hat{\mathbf{x}}_{k|k-1} \leftarrow \mathbf{A}\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}\mathbf{u}_k$                                  $\triangleright$  Project the state ahead
7:      $\mathbf{P}_{k|k-1} \leftarrow \mathbf{A}\mathbf{P}_{k-1|k-1}\mathbf{A}^\top + \mathbf{Q}$                        $\triangleright$  Project the error covariance ahead
8:
9:   Update Step
10:     $\mathbf{K}_k \leftarrow \mathbf{P}_{k|k-1}\mathbf{C}^\top(\mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}^\top + \mathbf{R})^{-1}$        $\triangleright$  Compute the Kalman gain
11:     $\hat{\mathbf{x}}_{k|k} \leftarrow \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \mathbf{C}\hat{\mathbf{x}}_{k|k-1})$            $\triangleright$  Update the state estimate
12:     $\mathbf{P}_{k|k} \leftarrow (\mathbf{I} - \mathbf{K}_k\mathbf{C})\mathbf{P}_{k|k-1}$                              $\triangleright$  Update the error covariance
13: end for

```

## 4.2 ソフトウェア評価

# 第 5 章

## 実環境におけるシステム評価

### 5.1 実験目的

(5.1) はガウス分布です.

$$f(x | \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) \quad (5.1)$$

### 5.2 実験環境

実験環境を説明します.

### 5.3 実験結果

ロボットのオドメトリは (5.2) で計算されます. 次の位置姿勢  $(x_{k+1}, y_{k+1}, \theta_{k+1})$  は, 現在の位置姿勢  $(x_k, y_k, \theta_k)$  と, 微小時間  $\Delta t$  の間の並進速度  $v$  および角速度  $\omega$  を使って更新されます.

$$\begin{cases} x_{k+1} = x_k + v\Delta t \cos(\theta_k + \frac{\omega\Delta t}{2}) \\ y_{k+1} = y_k + v\Delta t \sin(\theta_k + \frac{\omega\Delta t}{2}) \\ \theta_{k+1} = \theta_k + \omega\Delta t \end{cases} \quad (5.2)$$

#### 5.3.1 現在の位置姿勢

$(x_k, y_k, \theta_k)$  は現在の位置姿勢です.

### 5.3.2 次の位置姿勢

$(x_{k+1}, y_{k+1}, \theta_{k+1})$  次の位置姿勢です.

### 5.3.3 更新

微小時間  $\Delta t$  の間の並進速度  $v$  および角速度  $\omega$  を使って更新されます.

## 5.4 考察

考察します.

## 第6章

## シミュレーション評価

## 6.1 シミュレーション環境の構築

## 6.2 実験概要

6.3 考察

表 6.1 性能比較

| 項目        | 提案手法 | 従来手法 A | 従来手法 B |
|-----------|------|--------|--------|
| 精度 (%)    | 95.2 | 92.1   | 88.5   |
| 処理時間 (ms) | 12.5 | 18.0   | 10.2   |
| 消費電力 (W)  | 3.1  | 3.5    | 2.9    |

# 第7章

## 結言

### 7.1 結論

結論を述べます。

### 7.2 課題

課題を述べます。

### 7.3 展望

展望を述べます。

## 謝辞

共同研究者に感謝します。 いいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいい  
いいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいい  
いいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいい  
いいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいい

著者氏名

# 参考文献

- [1]国土交通省, **令和 6 年版国土交通白書**. 2024, pp. 141–142 (cit. on p. 1).
- [2]国土交通省, **第 25 回基本問題小委員会 配付資料：建設業を巡る現状と課題**, アクセス日：2025-9-17, 2023 (cit. on p. 1).
- [3]S. Halder and K. Afsari, “Robots in inspection and monitoring of buildings and infrastructure: A systematic review,” *Applied Sciences*, vol. 13, no. 4, p. 2304, 2023 (cit. on p. 1).
- [4]T. Sandakalum and M. H. Ang Jr, “Motion planning for mobile manipulators—a systematic review,” *Machines*, vol. 10, no. 2, p. 97, 2022 (cit. on p. 1).
- [5]総務省, **地中レーダー技術に関する調査検討会(平成 28 年度) 調査検討報告書：地中レーダーの現状**, アクセス日：2025-9-17, 2017 (cit. on p. 3).
- [6]佐藤源之, “地中レーダ (gpr) アンテナの電磁界的考察と計測法の提案,” **電子情報通信学会論文誌 B**, vol. 105, no. 11, pp. 844–851, 2022 (cit. on p. 3).
- [7]W. Chen, X. Li, et al., “Trajectory planning for spray painting robot based on point cloud slicing technique,” *Electronics*, vol. 9, no. 6, p. 908, 2020 (cit. on p. 3).
- [8]水内郁夫, 藤本純也, et al., “近接覚・触覚によるなぞり形状推定に基づく多種食器操作キッキンアシストシステムの実現,” **日本ロボット学会誌**, vol. 30, no. 9, pp. 889–898, 2012 (cit. on p. 3).
- [9]H. Rathod, S. Debeck, et al., “Applicability of gpr and a rebar detector to obtain rebar information of existing concrete structures,” *Case Studies in Construction Materials*, vol. 11, e00240, 2019 (cit. on p. 6).
- [10]Sensors & Software Inc. “Gpr の解像とターゲットの検出を理解する.” アクセス日：2025-9-17. [Online]. Available: <https://www.sensoft.ca/ja/blog/understanding-gpr-resolution-and-target-detection/> (cit. on p. 6).

付録 A

## 詳細な電子回路図



Fig. A.1 システム全体の回路図

## 付録 B

## 主要なプログラムのソースコード

## B.1 C++

**Listing B.1** Hello World Program in C++

```
// C++ source code here.  
#include <iostream>  
  
int main() {  
    std :: cout << "Hello ,\u25a1World!" << std :: endl ;  
  
    return 0;  
}
```

## B.2 Python

## **Listing B.2** Hello World Program in Python

```
# Python source code here.  
# The simplest way to print a line in Python:  
  
def main():  
    print( "Hello ,\u25a1World!" )
```

```
main()
```

# 付録 C

## 追加実験

### C.1 実験 A

ああああああああああああああああああああああああああああああああああああ  
ああああああああああああああああああああああああああああああああああああ

### C.2 実験 B

いいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいい  
いいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいい  
いいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいい