

視覚障がい者移動支援ロボット AI スーツケースの開発と 制御バリア関数を用いた衝突回避アシスト制御の実装

木村 駿介

(技術研究所)

Development of AI suitcase with control barrier function for collision avoidance

Shunsuke Kimura

視覚障がい者の移動を支援する自律型移動ロボットである「AI スーツケース」を開発している。視覚障がい者を目的地まで誘導する過程では、健常者の誘導や単体のロボットの移動と比較して、使用者やロボット自体が周囲の物体と衝突しない安全なロボット走行がより一層求められる。本稿では、清水建設で開発している AI スーツケースのコンセプトを紹介するとともに、衝突回避をアシストする制御バリア関数を用いた制御理論を説明する。提案法を実機に実装した結果から本理論の有効性を示す。

An autonomous mobile robot known as "AI Suitcase" that navigates visually impaired people to their destination is under development. To make it suitable for use by visually impaired people, AI Suitcase must avoid contact with building walls and fixtures as it performs its guidance task. In this paper, the control barrier function proposed to assist collision avoidance is explained. The effectiveness of this control theory is demonstrated using results obtained with an actual machine in which the proposed method is implemented.

1. はじめに

近年、高齢化に伴う視力の低下や緑内障をはじめとする目の疾患発症などにより、視覚障がい者は増加している。視覚障がいには様々な困難があるが、特に街を自由に移動できないことが社会参加の大きなハードルとなっている。例えば、経路を地図で確認して目的地に向かう、人や障害物を避けて通路を歩く、サインや看板を確認してお店に入る、駅やお店の列に並ぶ、知り合いを見つけて挨拶をするといった日常的な行動をとることが困難となる場面が多い。こうした課題を解決するため、清水建設は視覚障がい者のアクセシビリティと生活の質向上を目指す「一般社団法人次世代移動支援技術開発コンソーシアム」に参画して、スーツケース型ナビゲーションロボットである「AI スーツケース」の開発に取り組んでいる。AI スーツケースとは、カメラをはじめとした様々なセンサから得られる情報と、最新の AI・ロボット技術を組み合わせることで、視覚障がい者が自立して街を移動し自然なコミュニケーションを開始することを助ける統合ソリューションである。

技術的な観点から見た AI スーツケースの基本機能としては①位置情報と地図情報から目的地までの最適ルートを探索する機能、②音声や触覚などによ

る情報提示を交えながら誘導する機能、③映像およびセンサの情報から障害物を認識し避ける機能、の大きく分けて 3 つが挙げられる。これらの複数の機能を統合的に管理するために、AI スーツケースでは ROS (Robot Operating System) と呼ばれる、ロボットプラットフォームを用いて開発されている。

この AI スーツケースのオリジナルはカーネギーメロン大学にて開発された cabot¹⁾ と呼ばれるロボットであり、オープンソースハードウェア・オープンソースソフトウェアの思想に基づき、その情報は Github 上に公開されている²⁾。この cabot の思想がそのまま引き継がれてコンソーシアムでは AI スーツケースを開発している。またコンソーシアムの AI スーツケース開発と並列して、建物内での運用の際に求められるロボット機能を研究開発するために、清水建設でもこのリソースを活用して、図-1 に示す AI スーツケースを開発している。その理由としては、AI スーツケースの実用化に向けて、ガラスなどを含む建材や床材、大規模で複雑なレイアウトなど現実の建物内の状況へのこれまでには実現できていない新たな対応が必要となるからである。そのため、清水建設版 AI スーツケースは、追加のセンサやアクチュエータの着脱を可能とする拡張性に重きを置いており、新たな研究開発要素を取り入れた試験ができるように設計している。



図-1 清水建設版 AI スーツケース

しかしながら、新しい開発項目を検証する実験段階では、動作が不安定な場合もあり、またそのような状態で、ときには狭い場所での自律移動や手動操作が求められることもある。このように実験の過程では、壁や什器などとの接触を防止する機能が必要である。また、人や施工物への接触は、竣工後の建物にサービスロボットを導入するときだけに限らず、建築中の作業現場の作業ロボットなどにおいても大きな問題となる。この問題は古くより衝突回避問題として知られており^{3),4)}、近年でもなお盛んに研究されているテーマの一つである。

特に近年では、制御バリア関数(Control Barrier Function : 以降 CBF と記述する)による衝突回避問題への取り組みが研究されている^{5)~11)}。もともと CBF はシステムの状態が制約されたもとの目標値への安定化や追従を行う状態制約問題に対してよく用いられている。また、以上の方法のようにロボットの自律移動に着目された研究が多い一方で、中村らは拡張 CBF を提案して、それを用いて人間からの操作入力に対して安全性を保証する制御理論に関する研究として、ヒューマンアシスト制御を提案している^{9)~11)}。拡張 CBF の利点としては、他の CBF を使用した方法と比較して、複雑な関数の設計を要求としないことと、制約境界により漸近できることがあげられる。本研究では衝突回避を実現するために、拡張 CBF による衝突回避アシスト制御則を構築して、AI スーツケースに組み込んだ。

2. 拡張制御バリア関数

本章では、本稿で扱う拡張制御バリア関数について紹介する。詳細な証明については参考文献⁹⁾を参照されたい。はじめに、制御対象として入力アフィンな非線形制御システムとして

$$\dot{x} = f(x) + g(x)(u + u_h(t)) \quad (1)$$

を考える。ここで、 $x \in \mathbb{R}^{2n}$ はシステムの状態、 $u \in \mathbb{R}^m$ はシステムの入力を意味する。また、写像 $f: \mathbb{R}^{2n} \rightarrow \mathbb{R}^{2n}$ および $g: \mathbb{R}^{2n} \rightarrow \mathbb{R}^{2n \times m}$ はともに局所 Lipschitz 連続であると仮定する。一般的に状態 x の次元は n 次元で扱われるが、本研究では後々の整合性のため $2n$ 次元としている。また、一般的なシステムとは異なる点として、入力に連続写像 u_h が追加されており、これは人間の操作や自律制御信号などからの操作を意味している。 $u + u_h(t)$ をシステムへの入力と見れば、一般的な入力アフィンシステムとなり、 u により最終的なシステムへの入力を調整することができる。この理由から本稿ではこの u をアシスト制御入力と呼ぶ。

つぎに、システム(1) に対して拡張 CBF を以下で定義する。

定義. 1 (拡張 CBF)

システム(1) および安全集合 X を考える。ただし X は開集合で定義する。このとき以下を満たす C^1 級関数 B を拡張 CBF という。

(A1) B は非負の関数である。すなわちすべての $x \in X$ に対して $B(x) > 0$ 。

(A2) B はプロパー関数である。すなわち任意の正定数 $L \geq 0$ に対して、そのレベル集合 $\{x \mid B(x) \leq L\}$ はコンパクト集合。

(A3) 任意の $u_h(t)$ に対して以下を満たす $K \geq 0$ および $C \geq 0$ が存在する。

$$\inf_{u \in \mathbb{R}^m} \dot{B}(x, u_h(t), u) < KB(x) + C \quad (2)$$

ここで安全集合 X とは x の定義域である X と同等であり、本稿ではロボットの移動可能範囲を定義しているものと捉えることができる。定義. 1 の条件を満たしている拡張 CBF は以下の性質をもつ。

補題. 1 (拡張 CBF コントローラ)

システム(1)、安全集合 X 、および拡張 CBF である関数 B が与えられたとき、以下のアシスト制御入力 u を考える。

$$u = \begin{cases} 0 & (I(x, u_h) > J(x) \text{ or } L_g B(x) = 0) \\ \frac{I(x, u_h) - J(x)}{\|L_g B(x)\|^2} (L_g B(x)) & (\text{otherwise}) \end{cases} \quad (3)$$

このとき、このアシスト制御入力 u は任意の連続写像 $u_h(t)$ および任意の時間 t において $x \in X$ を保証する。ただし、 $I: \mathbb{R}^{2n} \times \mathbb{R}^m \rightarrow \mathbb{R}$ および $J: \mathbb{R}^{2n} \rightarrow \mathbb{R}$ は制御則の切り替えを判定する関数であり以下で定義される。

$$I(x, u_h) = L_f B(x) + L_g B(x) u_h(t) \quad (4)$$

$$J(x) = KB(x) + C \quad (5)$$

ここで、(5) の K および C は(2) と同一のパラメータである。また、 $L_f B$ と $L_g B$ は、ベクトル場 f と g のそれぞれの Lie 微分を意味しており、

$$L_f B(x) = \frac{\partial B}{\partial x}(x) f(x) \quad (6)$$

$$L_g B(x) = \frac{\partial B}{\partial x}(x) g(x) \quad (7)$$

で定義される。

証明.

参考文献⁹⁾ を参照されたい。 □

この拡張 CBF の性質を利用することで、ロボットのオペレーション中において常に安全集合内に状態を留めるようなアシスト入力 u を計算することができる。

総括すると、本制御法を実装するためには

- 1: 目的に応じた拡張 CBF である $B(x)$ を設計
- 2: I と J を計算
- 3: もし $I > J$ であれば u を計算

の3つの手続きが必要である。次章では、システム(1)にあたる本論文で扱う制御対象を明らかにして、そのシステムに対する拡張 CBF を提案する。

3. 機体座標系における拡張 CBF の設計

本論文では、機体座標系で観測した点群の運動に着目する。AI スーツケースには2次元の測域センサが搭載されており、周囲の物体との距離を角度に応じて取得できるとする。いま、測域センサが n 個の点を取得できたとして、 $i \in [1, n] \subset \mathbb{N}$ 番目の点に着目する。ここで \mathbb{N} は自然数全体の集合とする。機体座標系から見て角度 θ_i 方向に検出された物体との距離を r_i とすると、点の座標は $p_i = [r_i, \theta_i]^T$ と記述

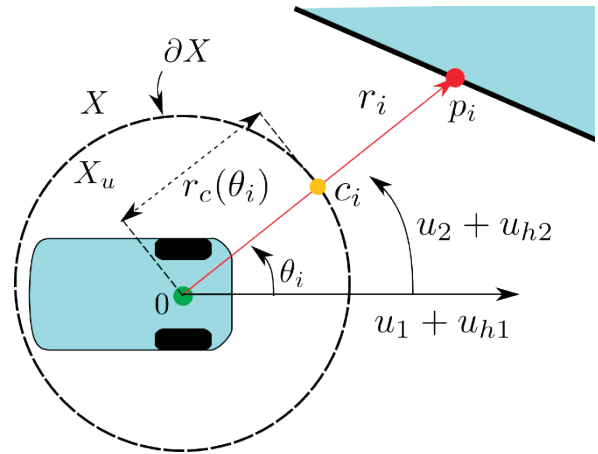


図-2 車両のモデル化

できる。これを図示したものが図-2 である。ここで、 $u_1 + u_{h1}$ はロボットの並進速度、 $u_2 + u_{h2}$ はロボットの回転速度を意味し、これらに応じて p_i は運動する。 p_i 、 r_i 、 θ_i 、 $u_1 + u_{h1}$ 、および $u_2 + u_{h2}$ 以外の文字は後ほど順を追って説明する。以上の設定のもと、以下の命題が成り立つ。

補題. 2 (点の状態方程式)

2 輪車両型移動ロボットに入力 $u + u_h$ が与えられたとき、点の座標 p_i を状態とした状態方程式は

$$\begin{aligned} \dot{p}_i &= \begin{bmatrix} -\cos \theta_i & 0 \\ \frac{\sin \theta_i}{r_i} & -1 \end{bmatrix} \begin{bmatrix} u_1 + u_{h1} \\ u_2 + u_{h2} \end{bmatrix} \\ &= g_i(p_i)(u + u_h) \end{aligned} \quad (8)$$

と記述される。

証明.

ロボットに並進速度 $u_1 + u_{h1}$ を入力するとき、その相対速度をベクトルの成分分解することで、点 p_i が持つ r_i 軸方向の速度は $-(u_1 + u_{h1}) \cos \theta_i$ と得られる。同様にロボットの回転速度 $u_2 + u_{h2}$ を考慮すると、 r_i 軸に対して垂直な方向の点 p_i の速度は $(u_1 + u_{h1}) \sin \theta_i - r_i(u_2 + u_{h2})$ と得られる。ここで r_i 軸に対して垂直な方向の点 p_i の速度を点 p_i の極座標形式のパラメータ θ_i で表すと $r_i \dot{\theta}_i$ であることに注意して、以上の結果を行列形式でまとめると式(8)が得られる。 □

つぎに、機体座標系 θ_i 方向における衝突回避を実現すべき距離として、図-2 に示す $r_c: S^1 \rightarrow \mathbb{R}_{\geq 0}$ を導入する。ここで、 S^1 は単位円周、 $\mathbb{R}_{\geq 0}$ は0以上

の実数の集合を意味しており、 r_c は θ_i に対して C^1 級関数であるとする。任意の i において、点 p_i が $r_i > r_c(\theta_i)$ の状態制約を満たす問題を考える。すなわち、点 p_i に対する安全集合を

$$P_i = \{p_i \mid r_i > r_c(\theta_i), \forall i \in N\} \quad (9)$$

とする。以上の設定のもとでヒューマンアシスト制御法を採用して $p_i \in P_i$ 、すなわち $r_i > r_c(\theta_i)$ を保証できれば、ロボットの機体は測域センサで θ_i 方向で検出された点には接触しない。この安全集合 P_i の状態制約を満たすための拡張 CBF の1 つとして以下の関数を提案する。

補題. 3 (点の状態方程式に対する CBF) 関数

$$B_i(p_i) = \frac{1}{r_i - r_c(\theta_i)} + L_0(r_i^2 + \theta_i^2) \quad (10)$$

はシステム(8) に対する拡張 CBF である。ただし、 $L_0 > 0$ は定数とする。

証明.

関数(10) に対して**定義. 1**の(A1)~(A3) の条件を確認していこう。衝突回避が達成されているとき、 r_i は $r_c(\theta_i)$ 以下の値を取り得ないため(A1) は明らか。(A2) の成立は(10) が下に凸の関数であることから言える。最後に(A3) については、**補題. 1** にて示した拡張 CBF コントローラ(3) が常に得られることから示される。□

この関数の要点は

$$\lim_{r_i \rightarrow r_c(\theta_i)} B_i(p_i) = \infty \quad (11)$$

であることである。また、関数(2) の第2項目は、(A2) を満たすために理論上必要であるが、実用上は正定数 L は0に近い値で適当に選ぶことができ、特にパラメータチューニングの必要はない。

以上ではひとつの点 p_i に着目して議論を進めてきたが、続いて測域センサでは n 個の点群が取得できるものとして議論を進める。 $i \in [1, n]$ に対して、点 p_i をすべてまとめた行列 $x = [p_1^T, \dots, p_i^T, \dots, p_n^T]^T$ を点群と呼ぶ。このとき点群の状態方程式は以下で表される。

命題. 1 (点群の状態方程式)

2 輪車両型移動ロボットに入力 $u + u_h$ が与えられたとき、点群 $x = [p_1^T, \dots, p_i^T, \dots, p_n^T]^T \in X \subset \mathbb{R}^{2n}$ を状態とした状態方程式を

$$\dot{x} = g(x)(u + u_h) \quad (12)$$

と記述するとき、状態方程式を構成する写像 $g(x)$ は

$$g(x) = \begin{bmatrix} g_1(p_1) \\ \vdots \\ g_i(p_i) \\ \vdots \\ g_n(p_n) \end{bmatrix} \quad (13)$$

である。

証明.

補題. 2 より得られた(8) の i に1から n まで代入した式を縦に並べた行列は

$$\begin{aligned} \begin{bmatrix} \frac{d}{dt} \begin{bmatrix} r_1 \\ \theta_1 \\ \vdots \\ r_i \\ \theta_i \\ \vdots \\ r_n \\ \theta_n \end{bmatrix} \\ \frac{d}{dt} \begin{bmatrix} r_1 \\ \theta_1 \\ \vdots \\ r_i \\ \theta_i \\ \vdots \\ r_n \\ \theta_n \end{bmatrix} \\ \frac{d}{dt} \begin{bmatrix} r_1 \\ \theta_1 \\ \vdots \\ r_i \\ \theta_i \\ \vdots \\ r_n \\ \theta_n \end{bmatrix} \end{bmatrix} &= \begin{bmatrix} g_1(p_1) \begin{bmatrix} u_1 + u_{h1} \\ u_2 + u_{h2} \\ \vdots \\ u_1 + u_{h1} \\ u_2 + u_{h2} \\ \vdots \\ u_1 + u_{h1} \\ u_2 + u_{h2} \end{bmatrix} \\ g_i(p_i) \begin{bmatrix} u_1 + u_{h1} \\ u_2 + u_{h2} \\ \vdots \\ u_1 + u_{h1} \\ u_2 + u_{h2} \\ \vdots \\ u_1 + u_{h1} \\ u_2 + u_{h2} \end{bmatrix} \\ g_n(p_n) \begin{bmatrix} u_1 + u_{h1} \\ u_2 + u_{h2} \\ \vdots \\ u_1 + u_{h1} \\ u_2 + u_{h2} \\ \vdots \\ u_1 + u_{h1} \\ u_2 + u_{h2} \end{bmatrix} \end{bmatrix} \quad (14) \\ \Leftrightarrow \frac{d}{dt} \begin{bmatrix} r_1 \\ \theta_1 \\ \vdots \\ r_i \\ \theta_i \\ \vdots \\ r_n \\ \theta_n \end{bmatrix} &= \begin{bmatrix} g_1(p_1) \\ \vdots \\ g_i(p_i) \\ \vdots \\ g_n(p_n) \end{bmatrix} \begin{bmatrix} u_1 + u_{h1} \\ u_2 + u_{h2} \end{bmatrix} \end{aligned}$$

と式変形できる。ここで点群 x を書き下すと

$x = [p_1^T, \dots, p_i^T, \dots, p_n^T]^T = [r_1, \theta_1, \dots, r_i, \theta_i, \dots, r_n, \theta_n]$ であるから、(12) の写像 $g(x)$ について(13) が成り立つ。□

このとき、システム(13) における安全集合は

$$X = \prod_{i=1}^n P_i \quad (15)$$

で記述される。ここで \prod は直和を表し、 $x \in X$ であるとき、任意の添数 i について $p_i \in P_i$ であることを意味する。式(15) で記述された安全集合 X に対して、

X の境界の点の集合を ∂X で表し、 ∂X よりもロボット側の領域は衝突判定エリア X_u と表現する。ここで衝突回避を実現すべき距離 $r_c(\theta_i)$ を用いて点 $c_i = [r_c(\theta_i), \theta_i]^T$ は任意の i で $c_i \in \partial X$ であるといえる。この様子を図-2 に示している。すなわち、すべての点 p_i からなる状態変数 x がつねに安全集合 X に存在することで計測された物体との接触が避けられる。これを実現するような、システム(12) に対して安全集合 X である拡張 CBF は以下で設計できる。

定理. 1 (点群の状態方程式に対する CBF)

関数(10) の総和から成る関数

$$B(x) = \sum_{i=1}^n B_i(p_i) \quad (16)$$

はシステム(12) に対する拡張 CBF である。

証明.

関数(16) に対して**定義. 1**の(A1)~(A3) の条件を確認する。(A1) は非負の関数の総和もまた非負の関数であるから明らか。(A2) は B_i が p_i 以外の点 (たとえば $p_j, j \neq i$) とは非依存であるためこれも成立する。(A3) についても**補題. 3** と同様に拡張 CBF コントローラ(3) が得られることから示せる。□

拡張 CBF を(16) と定義すると拡張 CBF コントローラ(3) の計算が容易という利点がある。具体的には、(4) の $I(x, u_n)$ を導出するために必要なシステム(12) のベクトル場 $g(x)$ の Lie 微分 $L_g B$ は、

$$\begin{aligned} L_g B(x) &= \frac{\partial B}{\partial x} g(x) \\ &= \left[\frac{\partial B_1}{\partial p_1}, \dots, \frac{\partial B_i}{\partial p_i}, \dots, \frac{\partial B_n}{\partial p_n} \right] \begin{bmatrix} g_1(p_1) \\ \vdots \\ g_i(p_i) \\ \vdots \\ g_n(p_n) \end{bmatrix} \quad (17) \\ &= \sum_{i=1}^n \frac{\partial B_i}{\partial p_i} g_{pi}(p_i) \end{aligned}$$

と得られる。これにより(4) の計算が容易に実装できる。さらに書き下すと、 B_i それぞれの計算結果を

$$\frac{\partial B_i}{\partial p_i} g_{pi}(p_i) = [L_{gp} B_{i1} \quad L_{gp} B_{i2}] \quad (18)$$

として、(8) と(10) から

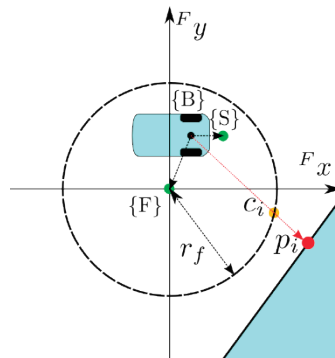


図-3 各フレームと点の位置

$$L_{gp} B_{i1} = - \left(- \frac{1}{(r_i - r_c(\theta_i))^2} + 2Lr_i \right) \cos \theta_i + \left(- \frac{\frac{dr_c(\theta_i)}{d\theta_i}(\theta_i)}{(r_i - r_c(\theta_i))^2} + 2L\theta_i \right) \frac{\sin \theta_i}{r_i} \quad (19)$$

$$L_{gp} B_{i2} = - \left(- \frac{\frac{dr_c(\theta_i)}{d\theta_i}(\theta_i)}{(r_i - r_c(\theta_i))^2} + 2L\theta_i \right) \quad (20)$$

と計算できる。またこれら(19) と(20) から r_c は引数 θ_i に対して C^1 級関数である必要性が確認できる。

まとめると、以上により、(4) と(5) より算出される I および J を比較して、ヒューマンアシスト制御則(3) が計算される。この制御則(3) により、状態 x は常に $x \in X$ が満たされ、移動ロボットと周囲の物体との衝突回避が実現できる。

4. センサ位置と使用者位置の考慮

前章の理論では、2 つの駆動輪の中心点と測域センサの位置が一致していることが要求されていたが、これはハードウェア構成を制限してしまうため実用的ではない。加えて AI スーツケースでは、ロボットの隣に使用者が存在する状態での運用が想定されているため、衝突判定エリアをロボットだけでなくヒトも含んだ円で設定したいというニーズもある。そこで、センサや衝突判定エリアの位置に自由度をもたせるため、フレームという概念を導入する¹⁴⁾。

図-3 に示すように、2 つの駆動輪の中心点を原点とする機体座標系をボディフレーム {B}、衝突判定エリアを記述する円の中心点を原点とする座標系をフットプリントフレーム {F}、最後に測域センサの設置中心位置を原点とした座標系をセンサフレーム {S} と表記する。これらのフレームの表記を利用して、あるフレーム上から見た座標を上付き文字で表

す。例えば ${}^S p_i$ はセンサフレーム上から計測した i 番目の点を意味する。同様に、 ξ で基準フレームから見た相対フレームの原点の位置を上付き文字と下付き文字を付した形式で表す。例えば ${}^B \xi_S$ はボディフレームから計測したセンサフレームの原点位置を意味する。本論文では簡単のため座標系の回転は考えない。そのため、 ξ は直交座標系 $[x, y]^T \in \mathbb{R}^2$ で定義でき、例として ${}^B \xi_S = [{}^B x_S, {}^B y_S]^T$ と表現する。

また、本章での ${}^S p_i$ すなわち前章で定義した $p_i = [r_i, \theta_i]^T$ は極座標系で記述されているため、直交座標系で表記した点はバー付き文字、例えば ${}^S \bar{p}_i = [{}^S x_{pi}, {}^S y_{pi}]^T$ で表す。極座標形式と直交座標系の間にはつぎの関係が成り立つことがよく知られている。

$${}^S \bar{p}_i = \begin{bmatrix} {}^S x_{pi} \\ {}^S y_{pi} \end{bmatrix} = \begin{bmatrix} r_i \cos \theta_i \\ r_i \sin \theta_i \end{bmatrix} \quad (21)$$

さらに、この逆写像は以下で得られる。

$${}^S p_i = \begin{bmatrix} r_i \\ \theta_i \end{bmatrix} = \begin{bmatrix} \sqrt{{}^S x_{pi}^2 + {}^S y_{pi}^2} \\ \text{atan2}({}^S y_{pi}, {}^S x_{pi}) \end{bmatrix} \quad (22)$$

ただし、ここで $\text{atan2} : \mathbb{R}^2 \setminus \{0\} \rightarrow (-\pi, \pi]$ は2引数アークタンジェントを意味しており、原点を除いて(21)と(22)は1対1対応している。互いに直交座標系であれば平行移動の演算が可能となり、相対座標について以下の系が成り立つ。

系. 1 (相対座標の計算)

相対座標を $\xi \in \mathbb{R}^2$ 、直交座標系で記述された一つの点を \bar{p}_i とする。いま、3つのフレーム $\{A\}$ 、 $\{B\}$ 、および $\{C\}$ を与えるとき以下の性質が成り立つ。

$${}^A \bar{p}_i = {}^A \xi_B + {}^B \bar{p}_i \quad (23)$$

$${}^A \xi_C = {}^A \xi_B + {}^B \xi_C \quad (24)$$

また、(23)と(24)より

$${}^A \bar{p}_i = {}^A \xi_B + {}^B \xi_C + {}^C \bar{p}_i \quad (25)$$

$${}^A \xi_B = -{}^B \xi_A \quad (26)$$

も自然に成り立つ。

つづいて上記より得られる各フレームとの相対位置を用いて、衝突回避を実現すべき距離 r_c を求める

方法について示す。まず、衝突判定エリアはフットプリントフレーム $\{F\}$ の原点を中心として半径 r_f の円で描かれるものとする。ここで、ボディフレーム $\{B\}$ の原点と測域センサにより計測される任意の点 p_i を通る直線を考え、この直線と衝突判定エリアを記述する円との交点は $c_i = [r_c(\theta_i), \theta_i]^T$ である。以上の位置関係を図示すると図-3 のようになる。

いま、 $\{B\}$ の原点から見た $\{F\}$ の原点は ${}^B \xi_F = [{}^B x_F, {}^B y_F]$ であるから、その極座標は(21)を用いて $[{}^B r_F, {}^B \theta_F]$ で記述される。以下に示す定理. 2 を用いると、 $\{B\}$ の原点から見た衝突判定エリアの境界 $r_c(\theta_i)$ を計算できる。

定理. 2 (衝突判定エリアの方程式)

ボディフレーム $\{B\}$ を基準座標系として考える。フットプリントフレーム $\{F\}$ の位置を極座標形式で $[{}^B r_F, {}^B \theta_F]$ として、その位置から半径 r_f で描かれた円を衝突判定エリアとする。ボディフレーム $\{B\}$ の原点から θ_i 方向の衝突判定エリアとの距離は

$$r_c(\theta_i) = \frac{{}^B r_F \cos(\theta_i - {}^B \theta_F)}{\sqrt{{}^B r_F^2 \cos^2(\theta_i - {}^B \theta_F) - {}^B r_F^2 + r_f^2}} \quad (27)$$

と計算される。

証明。

フレーム $\{B\}$ の原点から θ_i 方向の直線を考える。この直線と衝突判定エリアを記述する円との交点、 $\{B\}$ の原点、および $\{F\}$ の原点からなる三角形に対して余弦定理を用いると

$$r_f^2 = r_c^2 + {}^B r_F^2 - 2r_c {}^B r_F \cos(\theta_i - {}^B \theta_F) \quad (28)$$

が得られる。これを r_c について解くと(27)が得られる。厳密に言えば(28)は2次方程式であるため、その解は(27)の根号前の符号は正負の2通りが考えられる。しかしながら、幾何学的に

$$r_c(\theta_i) > {}^B r_F \cos(\theta_i - {}^B \theta_F) \quad (29)$$

を常に満たすため、符号は正と確定できる。□

また、(27)は θ_i に関して微分可能であり、その計算結果は

$$\frac{\partial r_c}{\partial \theta_i}(\theta_i) = -{}^B r_F \sin(\theta_i - {}^B \theta_F) - \frac{{}^B r_F^2 \sin(\theta_i - {}^B \theta_F) \cos(\theta_i - {}^B \theta_F)}{\sqrt{{}^B r_F^2 \cos^2(\theta_i - {}^B \theta_F) - {}^B r_F^2 + r_f^2}} \quad (30)$$

となる。

以上の結果を整理すると以下の手順で衝突回避アシスト制御則を計算できる。

- 1: 人間からの入力 u_h を計測
 - 2: 測域センサで点群 $p_i = [r_i, \theta_i]^T, \forall i$ を計測
 - 3: 式(27) より衝突までの距離 $r_c(\theta_i)$ を計算
 - 4: 式(10) と(16) より拡張 CBF $B(x)$ を計算
 - 5: 式(17) ~ (20)より Lie 微分 $L_g B(x)$ を計算
 - 6: 式(3) ~ (5) よりアシスト入力 u を計算
 - 7: 実機にシステム入力 $u + u_h(t)$ を入力
- 次章ではこれを実機に実装した結果を示す。

5. 実機実験

本理論の有効性の検証のために、AI スーツケースに実装して実験を行った。実験に使用した AI スーツケースはその正面に前方 270°の視野角をもつ 2D 測域センサを搭載しており、移動は前輪 2 輪が独立に駆動することで行う。センサの搭載位置は駆動輪の中心の位置から 0.15 m 前方に搭載されている。すなわち ${}^B \xi_S = [0.15, 0.00]^T$ となる。同様にフットプリントの位置も設定した。AI スーツケースのユーザはハンドルを左手で持ち、ロボットが進むべき方向を先導する形を取るため、フットプリントの位置はロボットのボディから見て右斜め後ろである ${}^B \xi_F = [-0.12, -0.25]^T$ と設定している。また、衝突判定エリアの半径はロボットとユーザを覆うように $r_f = 0.50$ m とした。衝突回避アシスト制御則を使用するにあたり、制御則のパラメータである(5) の K と C 、および(10) の L_0 も決定する必要がある。 L_0 については十分小さく選べば良いので 1.0^{-6} とした。今回と K と C については特にパラメータチューニングすることなく 1.0 と設定した。

以上のパラメータ設定のもとで、人間から壁に衝突するような制御入力を入れたときに、壁との衝突回避が実現できることを確かめた。ロボットは壁から 2.5 m ほど手前に設置して、人間からロボットへの速度入力 $u_h(t)$ は Joy スティックコントローラを用いて入力した。人間の速度入力は図-4 に示すように約 3 秒から並進速度 0.50 m/s を 15 秒時点まで入力し続け、ロボットを壁へと接近させた。その後、

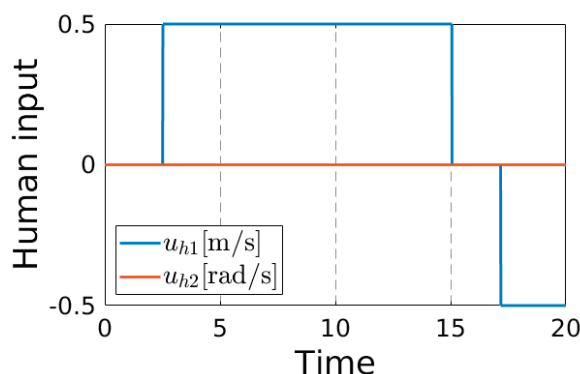


図-4 人間からの速度指令値の時間履歴

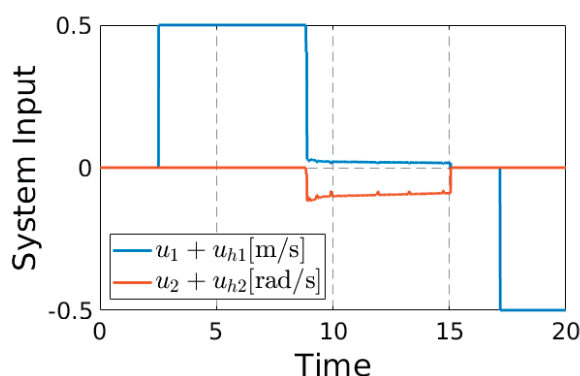


図-5 システムに与える指令値の時間履歴

17 秒時点から並進速度 -0.50 m/s を入力し、ロボットを壁から離す方向への速度指令を行った。人間の速度入力に制御則(3) からのアシスト入力を加えたシステムへの速度指令値の時間履歴を図-5 に示した。この図より、ロボットが持つ衝突判定エリアが壁へと到着する時間である 8 秒あたりからアシスト制御がかかり、システムへの並進速度入力が 0 に漸近していることがわかる。また同時に、できるだけ人間からの速度指令値を再現するために、並進速度は完全には 0 になっておらず、代わりに右回転方向の回転角速度を出していた。さらに壁から離れる方向の速度入力に関しては、アシスト入力は一切かからず、人間の操作を妨害しないことが確認できた。

つぎに、この実験結果をロボットの動きに関して考察する。ロボットの動きは機体座標系よりも絶対座標系で観察する方がわかりやすいため、ワールド座標系としてワールドフレーム $\{W\}$ を導入する。ロボットの初期位置を $\{W\}$ の原点として、ボディフレーム $\{B\}$ の位置 ${}^W \xi_B = [{}^W x_B, {}^W y_B]^T$ をロボットの位置とした移動軌跡を図-6 に示す。合わせてアシスト制御入力が機能している 15 秒時点での点群 x と計算された衝突判定エリア上の点群 $c \in \partial X$ を描画している。この図-6 から、衝突判定エリアの境界でロボットが停止できていることが確認できる。

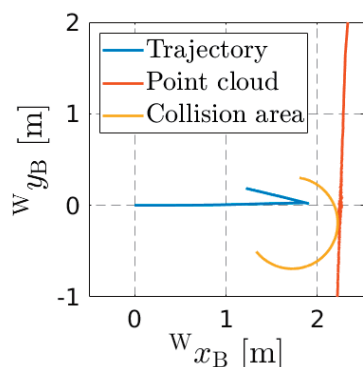


図-6 車両の軌跡と最終の衝突判定エリア

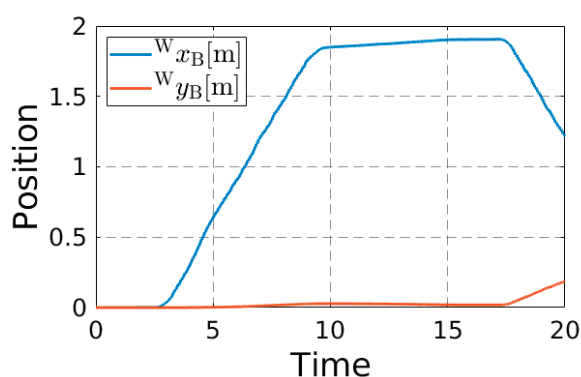


図-7 車両ボディフレームの位置の時間応答

また衝突判定エリアが点群に漸近したのちに、ロボットが回転動作をしたため、視野角 270° で計測されている衝突判定エリアが傾いている。姿勢角が変更された状態で17秒から負の並進速度を入力したために、 w_{yB} の値が増加している様子が図-7と合わせて確認できた。図-6の結果では、衝突判定エリアがAIスーツケースのユーザの立ち位置も含む形で考慮された衝突回避が実現できており、提案した拡張 CBF による制御則(3)の妥当性が示せた。

6. まとめ

本稿では、点群に着目した状態方程式とそのシステムに対する拡張 CBF を提案して、その妥当性を証明した。さらに、提案した拡張 CBF を用いて衝突回避アシスト制御則を設計して、視覚障がい者を施設内誘導するロボットである AI スーツケースに実装した。AI スーツケースは、2D 測域センサを車輪間中心からずらした位置に搭載しており、加えて、ユーザの立ち位置に応じて衝突判定エリアもロボット中心から変更する必要があった。提案制御則ではそれらのパラメータを任意に設定でき、実機実験の結果を通してその有効性を示した。拡張 CBF の利点とし

て、代数的な制御則のため反復計算などを解かせる必要はなく、制約境界にできるだけ漸近できた。さらに、衝突を発生させない方向への入力指令は邪魔せず、元の指令値を実行できることも確認できた。

<参考文献>

- 1) J. Guerreiro, D. Sato, S. Asakawa, H. Dong, K. M. Kitani, and C. Asakawa, "Cabot: Designing and evaluating an autonomous navigation robot for blind people," in the 21st International ACM SIGACCESS Conference on Computers and Accessibility, pp. 68–82, 2019.
- 2) <https://github.com/CMU-cabot/cabot>, 2021.11.12.
- 3) D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," IEEE Robotics & Automation Magazine, vol. 4, no. 1, pp. 23–33, 1997.
- 4) P. O'gren and N. E. Leonard, "A convergent dynamic window approach to obstacle avoidance," IEEE Trans. Robot., vol. 21, no. 2, pp. 188–195, 2005.
- 5) M. Z. Romdlony and B. Jayawardhana, "Stabilization with guaranteed safety using Control Lyapunov-barrier function," Automatica, vol. 66, pp. 39–47, 2016.
- 6) A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Barrier function based quadratic programs for safety critical systems," IEEE Trans. Autom. Control, vol. 62, no. 8, pp. 3861–3876, 2017.
- 7) Y. Chen, H. Peng, and J. Grizzle, "Obstacle avoidance for low-speed autonomous vehicles with barrier function," IEEE Trans. Control Syst. Technol., vol. 26, no. 1, pp. 194–206, 2018.
- 8) L. Lindemann and D. V. Dimarogonas, "Control barrier functions for signal temporal logic tasks," IEEE Control Systems Letters, vol. 3, no. 1, pp. 96–101, 2019.
- 9) 中村, 吉永, 小山, 江藤, "制御バリア関数を用いたヒューマンアシスト制御." 計測自動制御学会論文集 vol. 55, no. 5, pp. 353-361, 2019.
- 10) I. Tezuka and H. Nakamura, "Time-varying obstacle avoidance by using exact differentiator," in Proc. Anal. Conf. of the IEEE Industrial Electronics Society (IECON), pp. 571–576, 2019.
- 11) M. Igarashi, I. Tezuka, and H. Nakamura, "Time-varying control barrier function and its application to environment-adaptive human assist control," in Proc. IFAC Symp. Nonlin. Control Syst. (NOLCOS), vol. 52, no. 16, pp. 735–740, 2019.
- 12) P. Corke, Robotics, vision and control: "Fundamental algorithms in MATLAB® second, completely revised," Springer, vol. 118, 2017.