制御バリア関数を用いた 2 台のロボットの進路互譲

Distributed priority assignment for two robots based on mutual prediction via control barrier functions

システム工学群

人工知能システム論研究室　　川北　佳知

1. はじめに

制御分野における主要な研究対象の 1 つに，マルチロボット系がある．マルチロボット系の制御では，ロボット群が分散型のアルゴリズムによって協調し，グループ全体の目的を達成する．このような技術には，ロボット群による荷物の協調搬送や危険区域の探索といった応用が期待されている．

一方，近年では高価な高性能ロボットが一般向けに販売されつつある．例として，Boston Dynamics 社の四足歩行ロボット Spot[1]が知られている．高性能ロボットは単独で人間の代わりに作業ができるほどの性能を有しており，少子高齢化による労働力不足の解消に貢献するものと期待される．高性能ロボットの導入が進んだ社会では，機能も役割も異なる複数のロボットが同じ空間内に存在し，それらが独立してタスクを行いつつ，「必要なときだけ協調する」という状況になると予想される．この場合，全ロボットを常に協調させて 1 つのタスクを実行する従来のマルチロボット制御とは異なる制御が必要となる．

本研究では，こうした柔軟な協調が求められる例として，2 台のロボットによる進路の譲り合い（進路互譲）に焦点を当て，これを実現する制御法の開発を目的とする．進路互譲とは，衝突が予想される際に，互いの行動を予測して一方が進路を譲り他方が譲られることで，安全にすれ違うことを指す．進路互譲により最小限の動きで衝突を回避できれば，安全性の向上につながるとともに，無駄な動きを減らすことでタスクの効率化や省エネルギー化にもつながる．本研究では，制御バリア関数に基づく進路互譲の方法を提案し，その有効性をシミュレーションにより検証する．

記法：$\mathbb{R}$は実数の集合，$\mathbb{R}_+$は正の実数の集合を表す．ベクトル$\boldsymbol{x}$のユークリッドノルムを$\|\boldsymbol{x}\|$と表記する．

2. 問題設定

図 1 に示される，2 台のロボットからなる 2 次元空間上のマルチロボット系を考える．ロボットi $(i \in \{1,2\})$のダイナミクスは次式で与えられる．

$$\dot{x}_i(t) = u_i(t) \tag{1}$$

ここで，$x_i(t) \in \mathbb{R}^2$はロボットiの位置，$u_i(t) \in \mathbb{R}^2$は制御入力である．ロボットiには制御器

$$K_i: \begin{cases} u_i(t) = f_1\big(x(t), v_j(t), r_i\big) \\ v_i(t) = f_2(x(t), r_i) \end{cases} \tag{2}$$

が備えられているとする．ここで，$x(t) \coloneqq [x_1^{\mathsf{T}}(t)\ x_2^{\mathsf{T}}(t)]^{\mathsf{T}}$であり，$v_i(t), v_j(t) \in \mathbb{R}$はそれぞれロボット$i$と$j$ $(i \neq j)$の出力，$r_i(t) \in \mathbb{R}^2$はロボットiの目標位置，$f_1: \mathbb{R}^4 \times \mathbb{R} \times \mathbb{R}^2 \to \mathbb{R}^2$と$f_2: \mathbb{R}^4 \times \mathbb{R}^2 \to \mathbb{R}$は制御器を記述する関数である．

2 台のロボットには行動領域$\mathbb{Q} \subset \mathbb{R}^2$が設定される．$\mathbb{Q}$内には障害物等に相当する進入禁止領域$\mathbb{P} \subset \mathbb{Q}$が存在するとし，

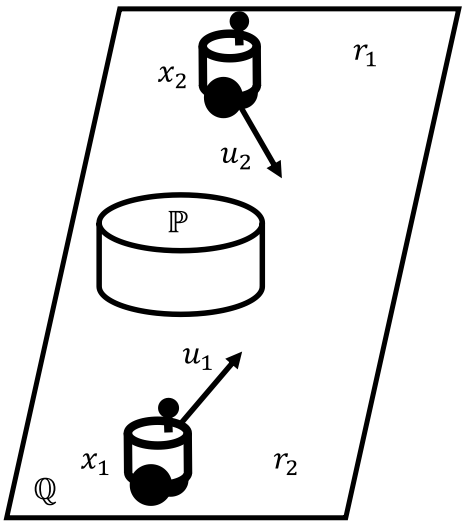


Fig. 1 Multi-robot system

ロボットはこの領域を通過できないとする．

このとき本研究では，与えられた目標位置$r_1, r_2 \in \mathbb{Q} \setminus \mathbb{P}$と任意の初期配置$x(0) \in (\mathbb{Q} \setminus \mathbb{P})^2$に対し，2 台のロボットが衝突することなく，

$$\lim_{t \to \infty} x_i(t) = r_i \quad \forall i \in \{1,2\} \tag{3}$$

を実現する制御器K_1, K_2（すなわち関数f_1, f_2）を求める問題を考える．ただし，ロボットが衝突を回避するときの移動量が小さくなるように，状況に応じて一方が他方の進路を確保するように振舞う，進路の譲り合いの実現を目標とする．

3. 制御バリア関数を用いた進路互譲

3.1. 制御バリア関数による制約付きマルチロボット制御

上記の問題を解くために，本研究では制御バリア関数に基づくアプローチ[2]を用いる．これは，システムが満たすべき制約を制御バリア関数に基づく不等式として記述し，その関数の性質を利用して制御入力を決定する方法である．

以下，図 1 のシステムを例として，文献[2]の方法の概要を説明する．図 1 のシステムに対し，ある関数$h(x)$を用いて，満たすべき制約を$h(x) \geq 0$と記述する．たとえば，2 台のロボットの間の距離を$d \in \mathbb{R}_+$以上に保つという制約は，

$$h(x) \coloneqq \|x_1 - x_2\|^2 - d^2 \tag{4}$$

に対し，$h\big(x(t)\big) \geq 0\ \forall t \in \mathbb{R}_+ \cup \{0\}$と表される．$h(x)$が図 1 の

システムに対する制御バリア関数であると仮定し，つぎの最適化問題を考える．

$$\min_{u_i \in \mathbb{R}^2} \|u_i - u_i^*\|^2 \quad \text{s.t.} \left(\frac{\partial h(x)}{\partial x_i}\right)^\top u_i \geq -w_i \alpha h(x) \qquad (5)$$

ここで，$i \in \{1,2\}$であり，$u_i^* \in \mathbb{R}^2$は制約がないときに(3)式を達成するノミナルな制御入力，$w_i \in \mathbb{R}_+$は$w_1 + w_2 = 1$を満たす重み，$\alpha \in \mathbb{R}_+$は定数である．最適化問題(5)式の目的関数はu_iとu_i^*の間の誤差を表す．一方，制約条件は$h(x)$が制御バリア関数であることから導かれ，各時刻tにおいてこれを満たすことで，$h\big(x(t)\big) \geq 0\ \forall t \in \mathbb{R}_+ \cup \{0\}$が成り立つ．ゆえに，各時刻$t$において(5)式を解き，得られた$u_i(t)$を用いることで，ロボット$i$は制約$h\big(x(t)\big) \geq 0\ \forall t \in \mathbb{R}_+ \cup \{0\}$を満たしつつ$u_i^*(t)$によるものに近い振舞いをする．

ロボット同士の距離と，各ロボットと領域$\mathbb{Q}, \mathbb{P}$の境界との距離を一定以上に保つという制約を考え，これを満たすように上記の方法を用いると，ロボット同士が衝突することなく$\mathbb{Q} \setminus \mathbb{P}$に留まることが保証される．このとき，ノミナル制御入力$u_i^*(t)$を適切に設計すれば，2 章の問題の解が得られると考えられる．しかし，この方法を用いると，初期配置$x(0)$によっては 2 台のロボットが同じように振舞ってしまい，進路互譲は実現されないことが確認された．

3. 2. 提案制御法

そこで本研究では，ロボット 1 と 2 で振舞いに差をつけるために，最適化問題(5)式における重みw_iに着目し，w_1とw_2を異なる値にすることを考える．$w_i, \alpha \in \mathbb{R}_+$であることと$h(x) \geq 0$であることから，$w_i$が小さいほど(5)における制約条件は厳しくなり，ロボットi $(i \in \{1,2\})$はより早い段階で衝突や進入禁止領域$\mathbb{P}$を回避するようになる．したがって，w_1とw_2を異なる値にすることは，一方のロボットが他方のロボットを優先的に行動させることを意味する．これにより，進路互譲が実現されると期待される．また，どちらのロボットを優先すべきかは状況によって異なるため，w_1とw_2の増減は状況に応じて決定することが望ましい．

このアイデアに基づき，w_i $(i \in \{1,2\})$を

$$w_i \coloneqq \begin{cases} 0.5 + \delta & \text{if } v_i < v_j \\ 0.5 & \text{if } v_i = v_j \\ 0.5 - \delta & \text{if } v_i > v_j \end{cases} \qquad (6)$$

$$v_i \coloneqq \left\|u_i^{opt} - u_i^*\right\|^2 \qquad (7)$$

と定める．ここで，$j \neq i$であり，$\delta \in \mathbb{R}_+$はw_iの変化量，$u_i^{opt} \in \mathbb{R}^2$は$w_i \coloneqq 0.5$のときの最適化問題(5)式の解である．(6)式のもとでは$w_1 + w_2 = 1$が成り立つことに注意する．まず$w_i \coloneqq 0.5$として(5)式を解き，その解u_i^{opt}に基づいて，ロボットiは(7)式のv_iをロボットjへ送る．そして，送信されたv_jとv_iの大きさを比較しw_iを増減させる．$v_i < v_j$のとき，ロボットiの方がノミナル制御入力u_i^*からu_i^{opt}への修正量が少ないという意味で(5)式における制約条件を容易に満たせるため，w_iを増加させて自身が優先的に振舞う．逆に，$v_i > v_j$のときはw_iを減少させて，ロボットjが優先的に振舞うようにする．本研究では，各時刻tにおいて，以上の方法により決定したw_iに対し(5)式を解き，得られた$u_i(t)$を用いることを提案する．

提案制御法の有効性をシミュレーションにより検証する．図 1 のシステムを考える．ロボットi $(i \in \{1,2\})$に対するノミナル制御入力$u_i^*(t)$を

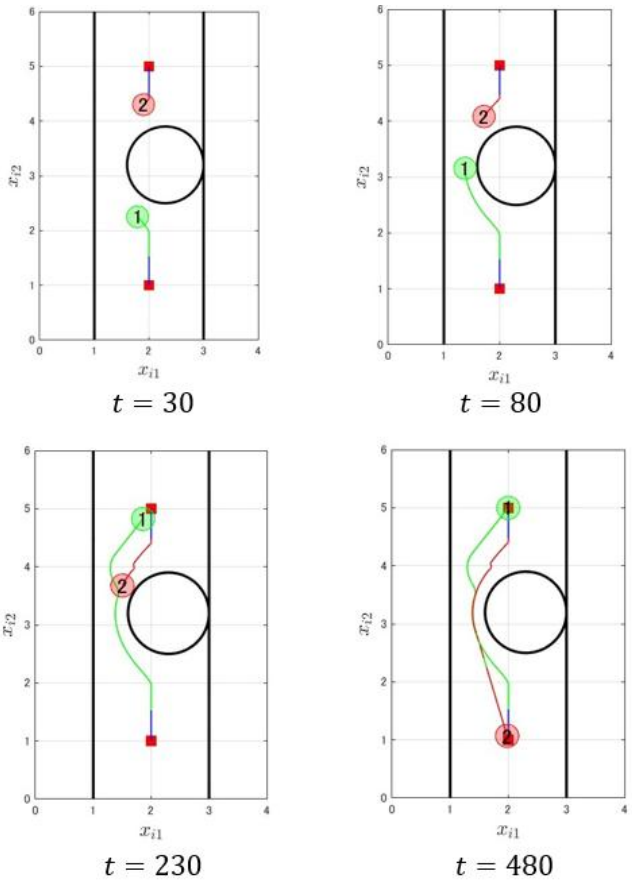


Fig. 2 Time series of robot positions

$$u_i^*(t) = -2(x_i(t) - r_i) \qquad (8)$$

とする．ロボット同士が保つべき距離を$d \coloneqq 0.4$とし，(5), (6)式におけるパラメータを$\alpha \coloneqq 3$, $\delta \coloneqq 0.3$とする．

このときのシミュレーション結果を図 2 に示す．ここで，中央の円が進入禁止領域$\mathbb{P}$，2 つの●がロボット 1 と 2，対応する■が目標位置r_1, r_2を表す．ロボット 1 が優先されて先にr_1へ向かった後，ロボット 2 がr_2へ向かう様子が観察できる．

4. おわりに

本研究では，2 台のロボットを対象に，進路の譲り合いを実現する問題を検討した．制御バリア関数を利用して制約を満たすアプローチに着目し，状況に応じてロボットごとに制約条件に対する重みを変えるというアイデアによって進路の譲り合いを実現する方法を提案した．そして，提案制御法の有効性をシミュレーションにより示した．

参考文献

(1) Boston Dynamics: Spot–The Agile Mobile Robot, https://bostondynamics.com/products/spot/, accessed 30 Oct. 2025

(2) 船田，山内，畑中：制御バリア関数に基づく制約付きマルチロボット制御，システム/制御/情報, Vol. 69, No. 2, pp. 41–46 (2025)