

# Appendix 2 Contents

Robocode の高等技術 - パターンマッチング - ..... 1

パターンマッチング	1
パターンマッチングの原理	1
ソースコード	3
パターンマッチングの効果	3

## Appendix 2

# Robocode の高等技術 - パターンマッチング -

このドキュメントでは Robocode の高度なテクニックについて解説しています。

ここでは、複雑な動きをする敵ロボットの動きを予測するパターンマッチングについて解説します。

このサンプルロボットは、STAGE 5 で解説した AdvancedRobot クラスの Droideka に組み込みます。

## パターンマッチング

パターンマッチングは敵の動きを記録して、そこから現在の動きに合ったパターンを探すことでの次の動きを予測します。そのために、直線・円形・往復運動するロボットだけでなく、複雑な動きを見せるロボットにも効果があります。

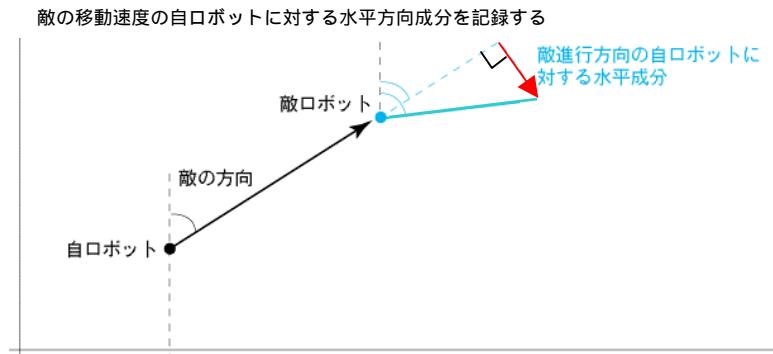
ここでは、Robo Wiki (<http://robowiki.dyndns.org/>) の解説（英語）を参考にしました。

## パターンマッチングの原理

パターンマッチングでは、敵の動きを長時間に渡って記録します。すべての情報ではなく、敵の移動速度の自ロボットに対する水平方向成分を記録します。

### 敵の移動速度（自ロボットに対する水平方向成分）

$$= \text{敵ロボットの速度} * \sin(\text{敵の方向} - \text{敵の進行方向})$$



記録には、**配列**と **StringBuffer クラス**を使います。

配列とは、変数に番号を付けたもので、その番号で特定の変数の位置を指定できます。StringBuffer クラスは、Java クラスライブラリに用意された標準部品の 1 つです。配列とよく似ていますが、検索用のメソッドを最初から持っています。本来は文字列を格納するクラスですが、データを char 型に変換することで他のデータ型を保存できます。

パターンマッチングでは、敵口ポットの移動速度を StringBuffer クラスに格納し、その変化分を配列に格納します。

**敵口ポットの移動速度** ----- patternMatcher クラス

**敵口ポットの移動速度の変化分** ----- arcLength 配列

パターンマッチングでは、記録されている敵口ポットの動きから最近の動きと同じ部分を探し、それに続く動きを移動予測に利用します。

#### *Column* --- StringBuffer クラスのメソッド

パターンマッチングでは、"StringBuffer" クラスが持つ次の 3 つのメソッドを使います。

**append() メソッド** ----- StringBuffer クラスにデータを追加する

**substring() メソッド** ----- StringBuffer クラスからデータを取り出す

**lastIndexOf() メソッド** ----- StringBuffer クラスから特定のデータの位置を検索する

StringBuffer クラスの使い方は、次のサイトに掲載されています。

- Java 2 プラットフォーム API 仕様

<http://java.sun.com/j2se/1.4/ja/docs/ja/api/index.html>

## ソースコード

Michael Dorgan が開発した Moebius のコードを参考にして、パターンマッチングのロボットプログラムを作りました。このロボットはパターンマッチングを採用した短いロボットプログラムで、Minibot Challenge で 1 位になったこともある実力派ロボットです。

Appendix 2 で登場したロボットのサンプルコードは、付録 CD-ROM の次のフォルダに収録してあります。

- ・付録 CD-ROM の「sample\_code」 「Appendix2」 フォルダ

- ・ファイルの内容

Droideka_Pattern.java	-----	パターンマッチングを実装した Droideka
Enemy.java	-----	敵情報を管理するクラス
Direction.java	-----	方向管理クラス
Lib.java	-----	共通処理クラス

## パターンマッチングの効果

doAim() メソッドを組み込んだロボットは、直線予測や円形予測がなくても確実に敵を攻撃できます。ただし、Droideka に組み込むにはいくつか調整が必要です。自分の位置が動きすぎると、予測機能が十分に効果を出せないことがあるようです。