

ロボット博士養成講座

ロボティクスプロフェッサーコース

とうりつしんし

倒立振子ロボット③

第6回

とうりつしんし

倒立振子ロボットを動かして遊ぶ

講師用

目 次

0. 倒立振子ロボットを動かして遊ぶ

0.0. 「倒立振子ロボットを動かして遊ぶ」でやること

0.1. 必要なもの

0.2. 倒立振子ロボットの準備

1. 倒立振子ロボットの制御方法

1.0. 前回のおさらい

1.1. 姿勢検出シールドのセンサーの活用

1.2. PID 制御

2. 倒立振子ロボットを動かす

2.0. 倒立振子ロボットの調整方法

2.1. 調整しながら動かしてみる

3. まとめ

○ 授業開始にあたって

授業のはじめは、着席させ、大きな声であいさつしてから始めます。

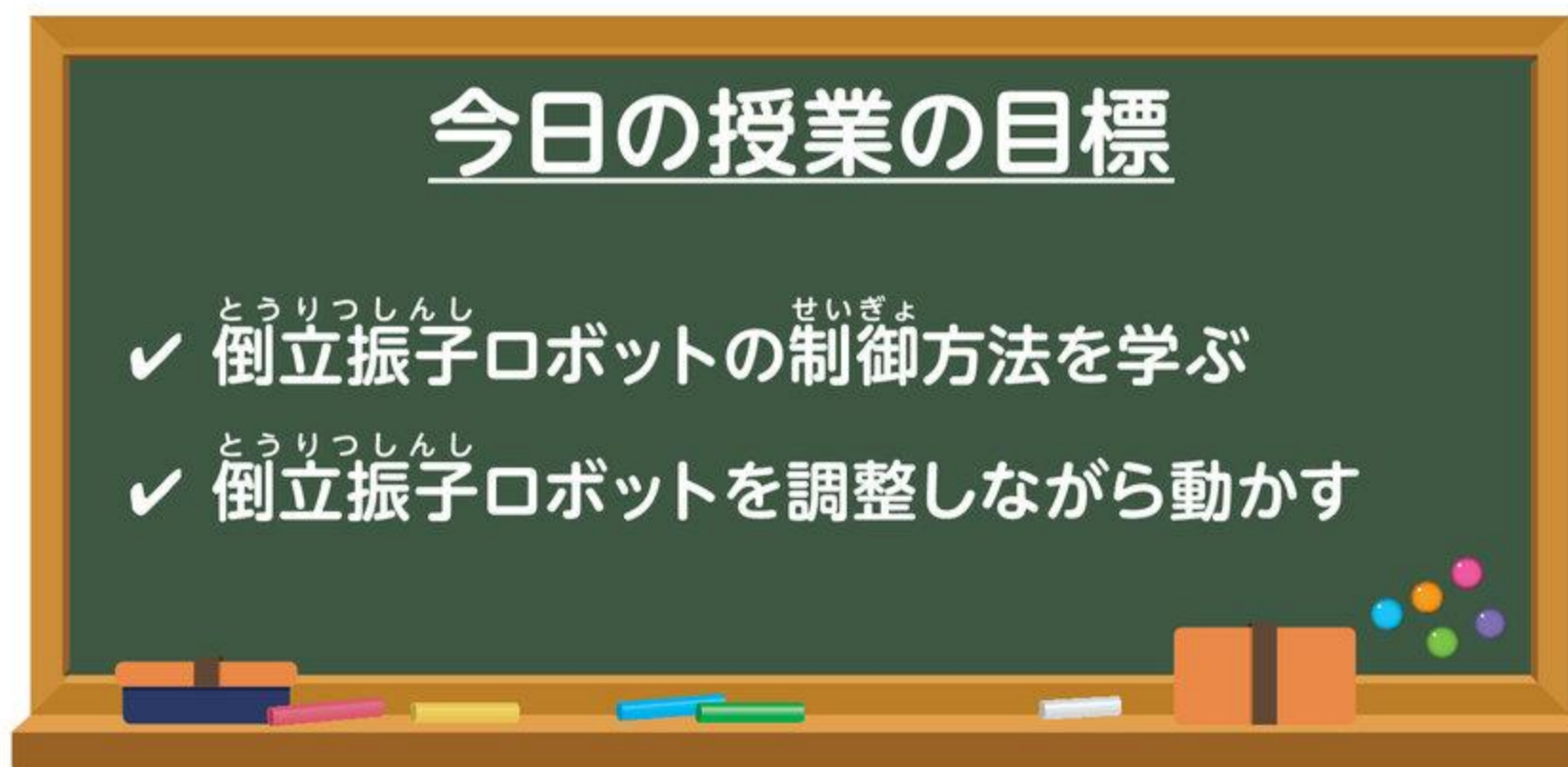
○ 今回の目標をパネルで用意するか、黒板に予め書いておきます。

(授業の目標を明確化することは大変重要なことですので、生徒によく理解させます)

目安時間は授業時間 120 分のうち、休憩 10 分程度取ることを想定しています。
生徒の進捗状況により、休憩時間などを調整して授業を行ってください。

0. 倒立振り子ロボットを動かして遊ぶ (目安 10分)

0.0. 「倒立振り子ロボットを動かして遊ぶ」でやること



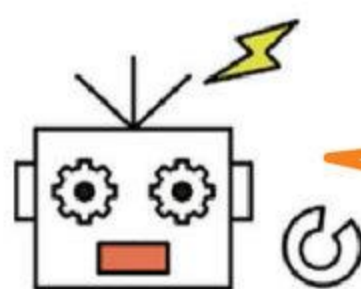
第6回では、前回あまり踏み込まなかった「倒立振り子ロボット」の制御方法について、勉強していきます！ 完成させたロボットを見ると、タイヤが2個しかないため、やはり制御方法がカギになります。

一見難しそうな動作を実行させるのが、ロボットの制御の醍醐味です。

人間でも一輪車で移動するのは難しいですね。それと同じようなことをどうロボットにさせるかを今回は勉強します。数学の複雑な部分はあまり説明しませんが、制御方法と調整方法をしっかりと身につけましょう。ロボットの調整がそのまま性能に出ると思います。では、はり切っていってみましょう！



図 0-0 いちりんしゃのイメージ図



倒立振り子ロボットの凄さを見せてやるぞー！

0.1. 必要なもの

前回作ったロボットと、以下のパーツを準備しておきましょう。

USB ケーブル	1	コントローラー	1
			

図 0-1 必要なもの

0.2. 倒立振子ロボットの準備

前回は可変抵抗ボリュームのテストのために7セグメントLEDを使用しましたが、今回は取り外しておきます。理由としては、倒立振子ロボットはとても繊細な電気信号を姿勢検出シールドから受け取りますが、7セグメントLEDを使用すると微弱なノイズが発生してロボットの制御に影響を与えるからです。

ということで、まずは7セグメントLEDとマトリクスLEDシールドを取り外します。

また、電池残量が少なくなるとロボットの動作に差が出るので、交換用の電池も用意しておくといよいでしょう。

さらに、ネジの緩みや、各シールドなどの接続、回路の配線も確認しておきましょう。もし切れかかっているジャンパー線があれば交換しておきましょう。

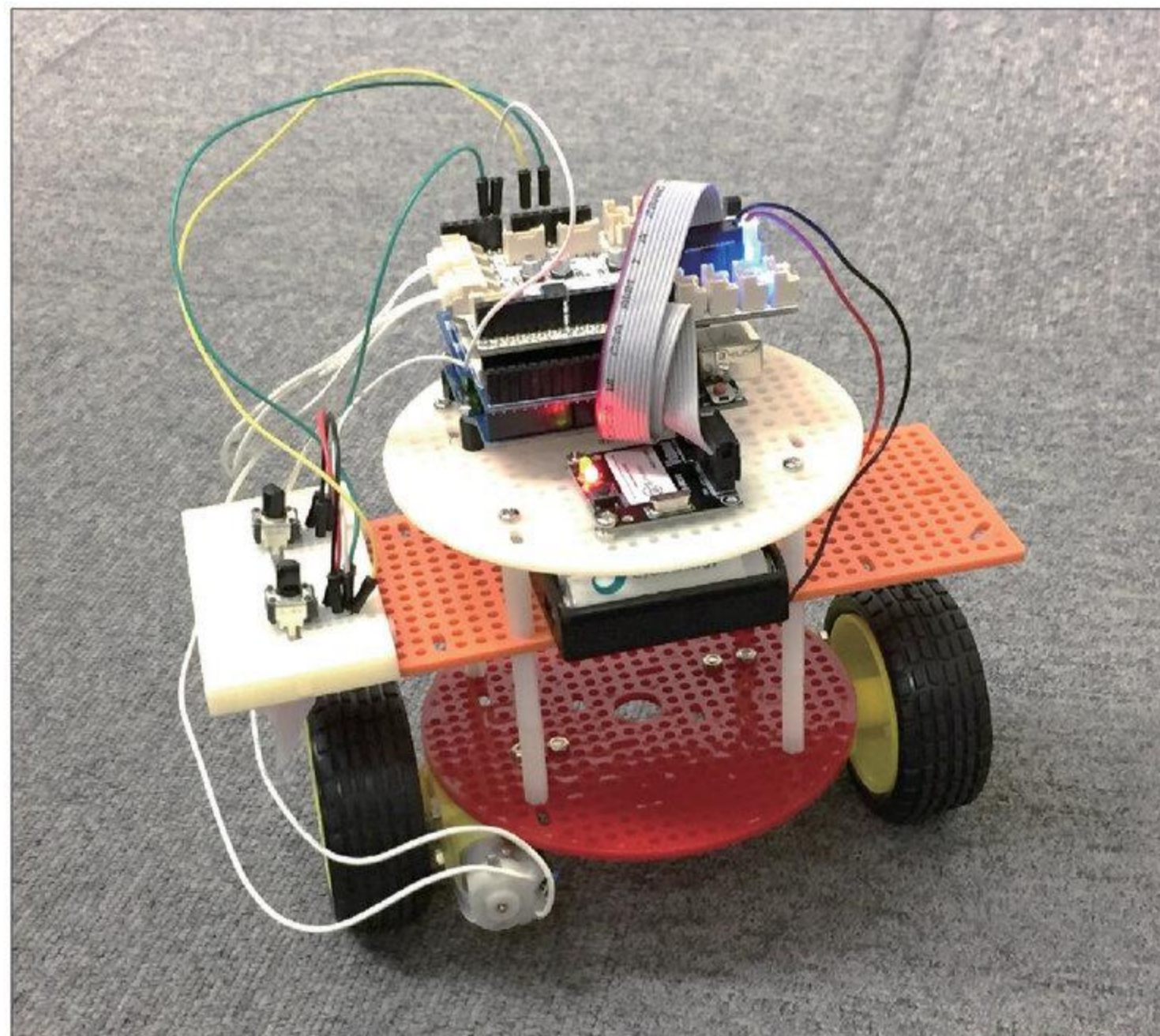


図 0-2 倒立振子ロボット

1. 倒立振り子ロボットの制御方法 (目安 35分)

1.0. 前回のおさらい

第5回では、実験により、支点から重心までの距離が遠いものの方が安定しやすいなどということがわかりました。理由としては、「支点から重心までの距離が大きいほど、物体が傾いたときに立て直しのチャンスが多くあるため」でした。

さらに、「慣性の法則」を用いた姿勢を保つ仕組みを学びました。物体の傾きの方向と同じ方向に支点を移動させることによって、重心の位置を復元するという方法でしたね。

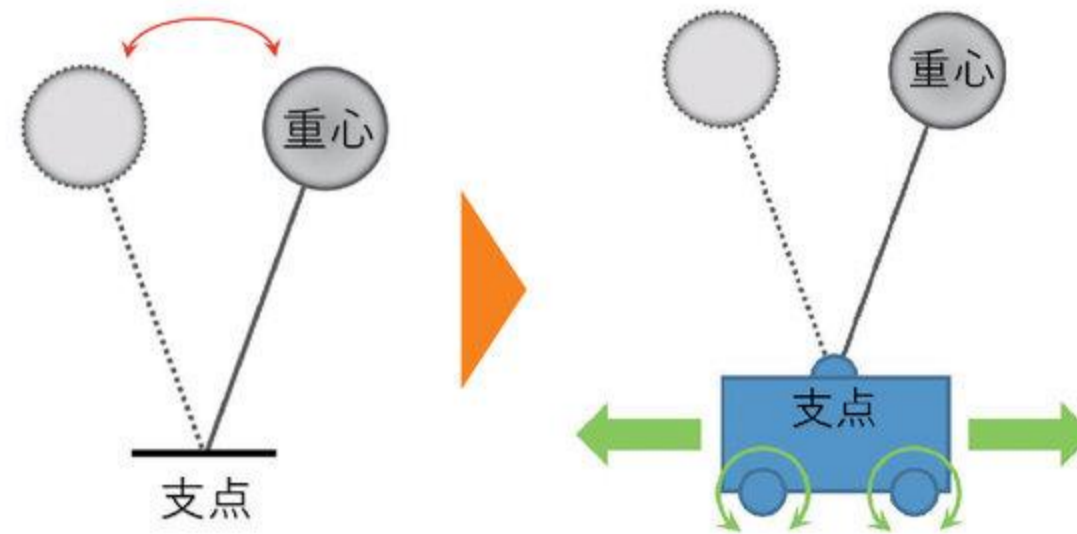


図 1-0 倒立振り子の仕組み

これをロボットで利用する場合、モーターの位置や速度を目標値に制御する必要があります。そのために、姿勢検出シールドのセンサーで現在の位置や速度などを検出し、その情報をもとにモーターの動作を加減する「フィードバック制御」を行います。たとえば、一輪車を乗りこなすときに、人間が感覚的にやっているフィードバックと同じです。



POINT

一輪車におけるフィードバック

- ①自分の状態を感じる (対象の状態を見る)
- ②ふらついたときに安定姿勢まで戻そうとする (目標との差を見る)
- ③ペダルへの力加減や体のバランスを調整する (出力を変化させる)



図 1-1 一輪車の例

1.1. 姿勢検出シールドのセンサーの活用

倒立振り子ロボットでは、姿勢を保つためにセンサーで角速度（回転する速さ）を検出します。図1-2のように、ロボットが前方後方に倒れる速度を検出して、フィードバックします。このときロボットは、倒れる方向と同一方向に移動し、倒れる速度によって移動速度を調整しています。

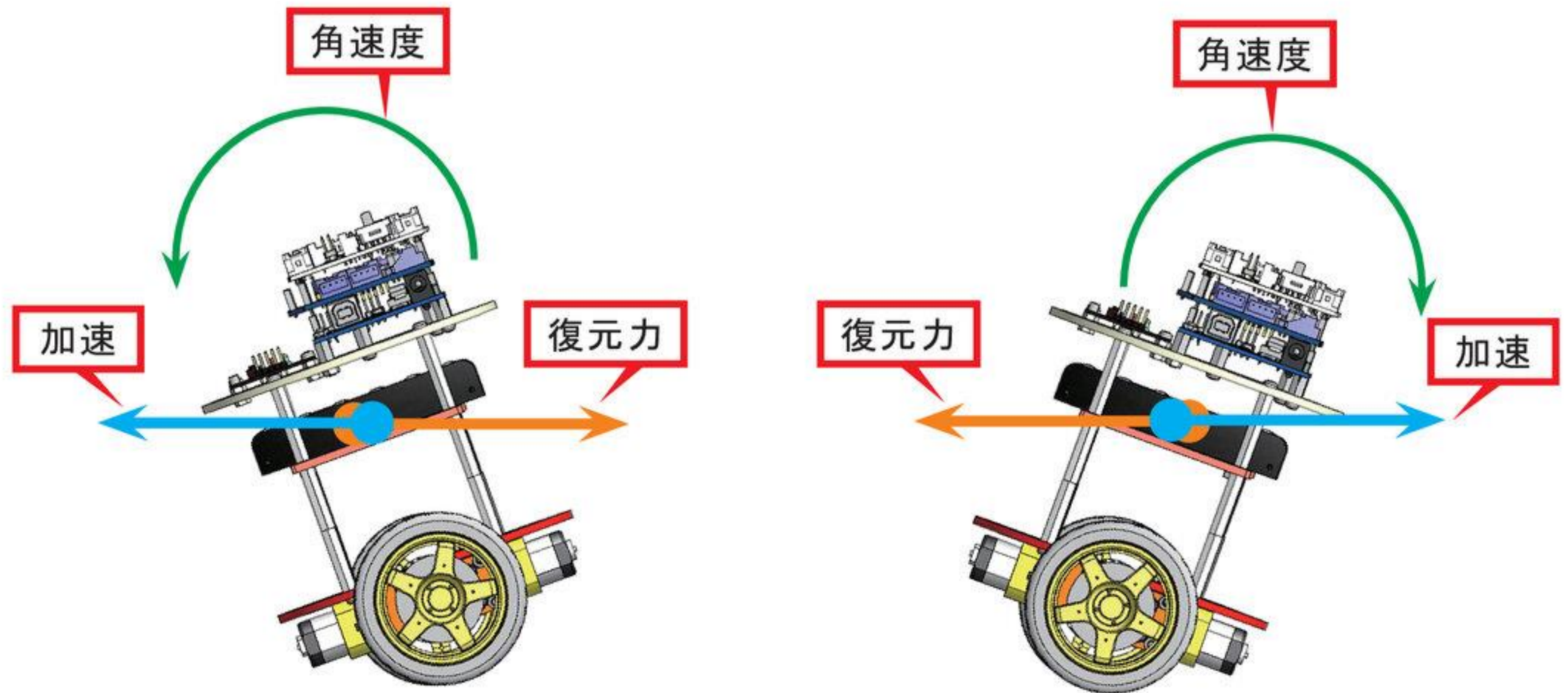


図1-2 フィードバック制御

1.2. PID 制御

1) 比例制御 (P 制御)

以前の回で、ロボットや機械に、目標とのずれ（偏差）に比例した出力を行わせる「比例制御 (P 制御)」について学びました。

これは偏差が倍になったら出力も倍にして動作させることで、より効率的に目標に近づけるという制御方法でしたね。

比例制御は考え方も数式も実にシンプルなのですが、同時に以下のような問題も抱えています。

POINT

- ① 環境や状況によっては、制御したあとも偏差が残ってしまうことがある。
- ② 目標に勢いよく近づきすぎて反対側に飛び出てしまうことがある。

このような問題に対処するため、比例制御に加えて次のような処理を行うことがあります。

2) 積分制御 (I 制御)

①の問題は、さまざまな原因で起こります。

たとえば、とある空っぽの冷蔵庫の目標温度が3度で、実際の庫内の温度が10度だったとしましょう。偏差があるので、冷蔵庫は冷却装置を動かして温度を下げようとするはずですが、同じ温度・同じ冷蔵庫でも中に物がぎゅうぎゅうに詰め込まれていたらどうでしょうか。空っぽのときと同じ出力で冷やそうとしたら、恐らく3度まで冷やすことはできないはずですが、ひょっとしたら、5度くらいで温度低下が止まってしまうかもしれませんね。このように、狙い通りの出力で制御を行ったのに、何らかの原因で結局残ってしまった偏差のことを定常偏差（オフセット）といいます。

倒立振り子ロボットだと、たとえばモーターは回転させているものの、ツルツルした床でタイヤが空回りしてしまって姿勢を戻しきれない、などといった状況がありえますね。

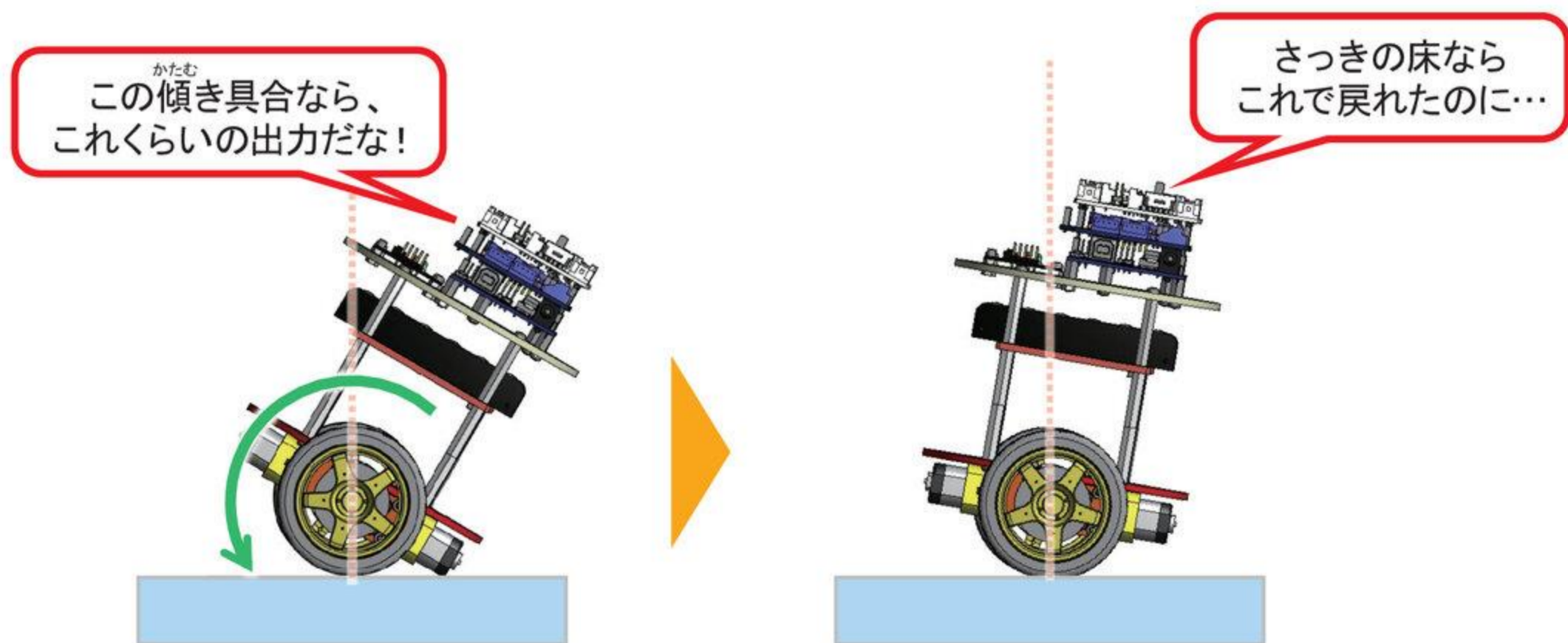


図 1-3 倒立振り子ロボットの定常偏差

定常偏差を打ち消すためには、「今現在の偏差」だけでなく「これまでの偏差の傾向」に目を向ける必要があります。

たとえば「きちんと比例制御をしているはずなのに、なぜかさっきから後ろに傾いている瞬間が多いなあ」という状況だったら、少し前に倒すような出力値にしてあげればよいわけです。

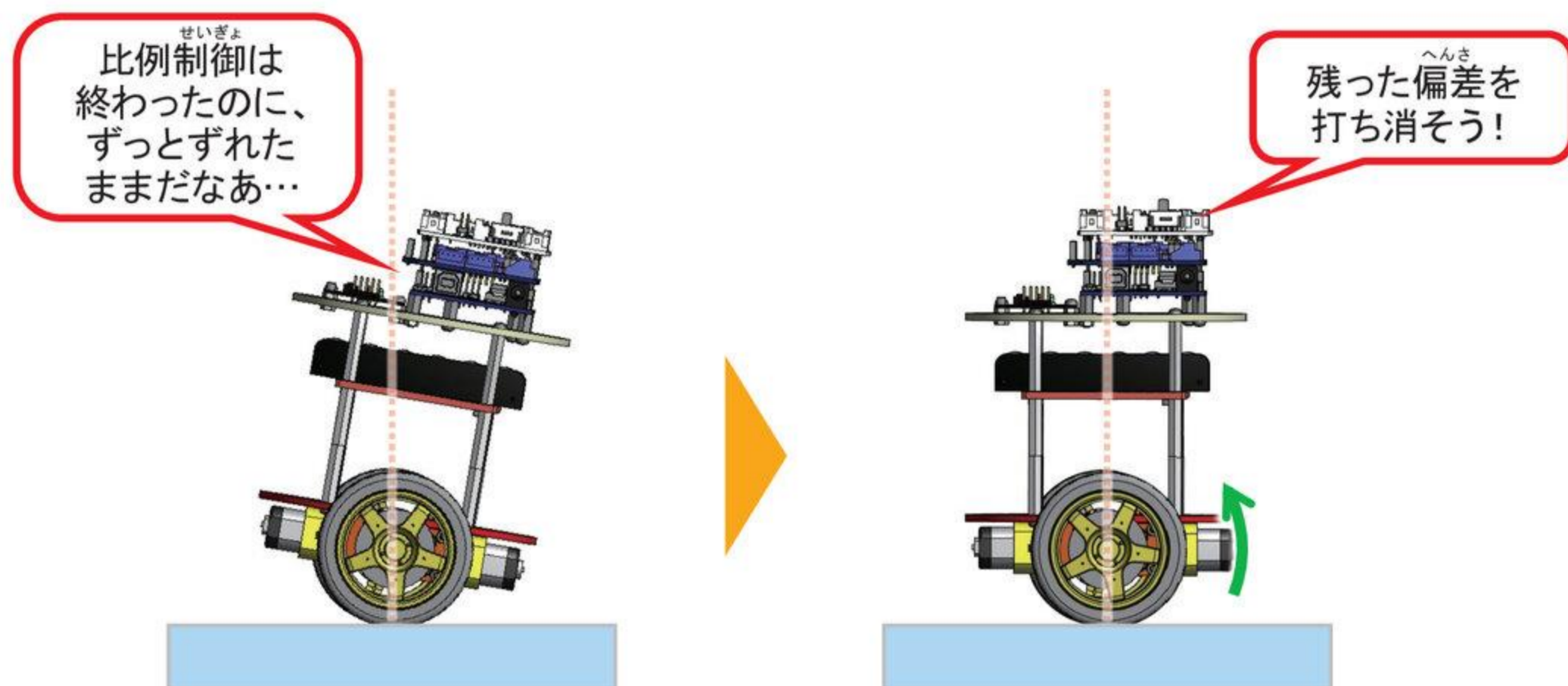


図 1-4 倒立振り子ロボットの定常偏差を抑える

このように「これまでの偏差の和（合計）から偏差の傾向をわりだし、それを打ち消すように出力値を変更する」という制御方法を「積分制御」といいます。「積分」はもう少し先の学年で学ぶ数学の考え方で、英語では「integral」と書きます。そのため、頭文字をとって「I制御」とよぶこともあります。

3) 微分制御（D制御）

倒立振り子ロボットは「倒れる前に出力値を調整する」必要があるのですが、比例ゲインを高くしてすぐにモーターが回転するようにしたいところですが、そうすると②のような問題が起こってしまいます。

このように、勢いあまって目標値を乗り越えてしまうことを「オーバーシュート」といいます。

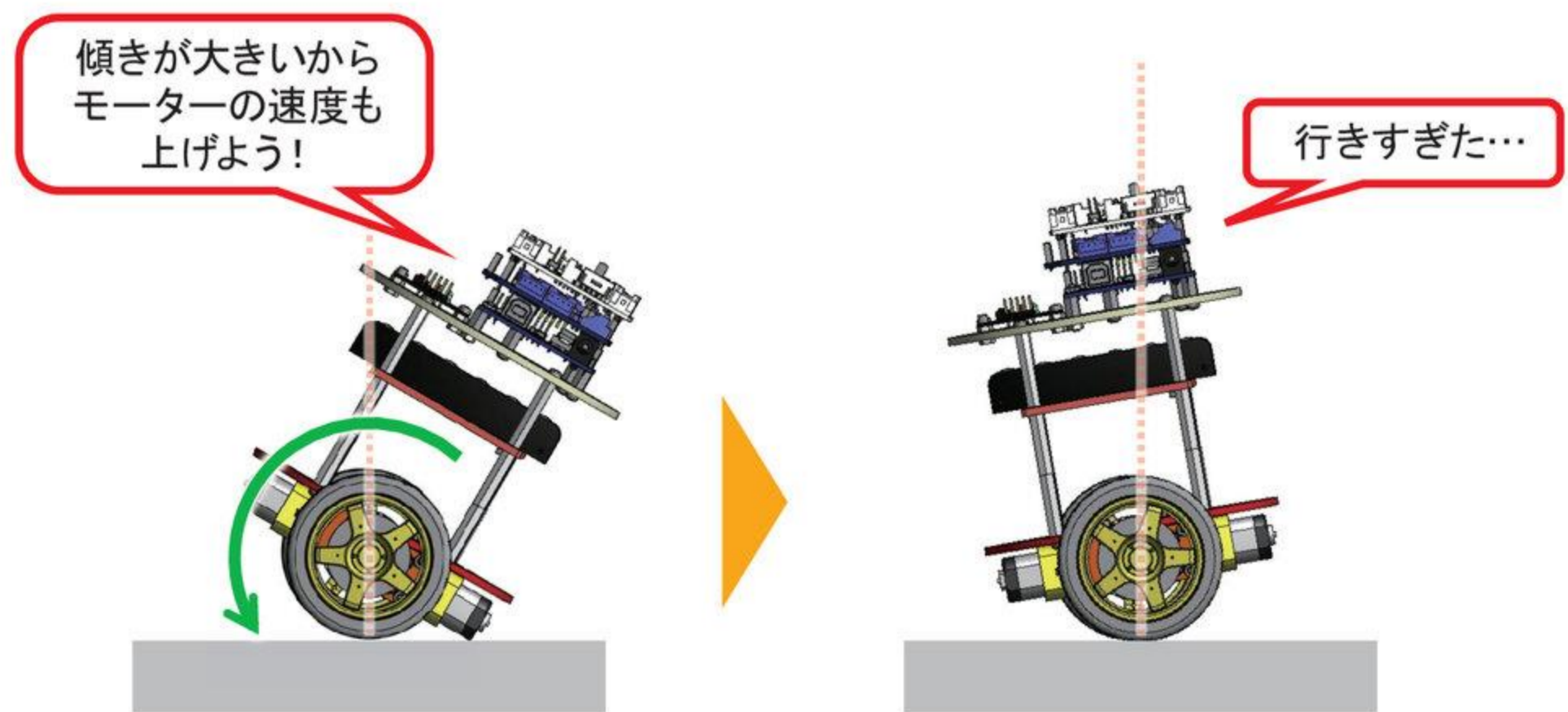


図1-5 倒立振り子ロボットのオーバーシュート

先ほど紹介した積分制御は最終的な出力値を調整するには役立ちますが、ある程度定常偏差のある時間が続かないと最適な出力値を求められません。つまり、出力値をイイ具合に直すのに時間がかかるのです。

皆さんも自転車で急な下り坂にさしかかったとき、スピードが一気に上がったなら「まずい、このままだと下で止まれなくなるぞ」と危機感を覚えるはず（覚えてくださいね！）。ここでのんきに積分制御なんてしていたら、答えを求める前に坂を駆け下りきってクラッシュしてしまいます。

これではオーバーシュートの問題に対処するのは難しいので、もう1つちがった制御方法を取り入れてみます。

オーバーシュートの原因は要するに「勢いよく目標に近づきすぎる」ことなので、勢いがよすぎるときは「まずい、このままだとオーバーシュートするぞ」とロボットが危機感を覚えてくれればよいわけです。

つまり、ロボットが「今現在の、目標へ向かう勢い」を判定し、その勢いに応じて出力値を調整してくれるようにすれば、オーバーシュートを抑えることができるようになるということです。

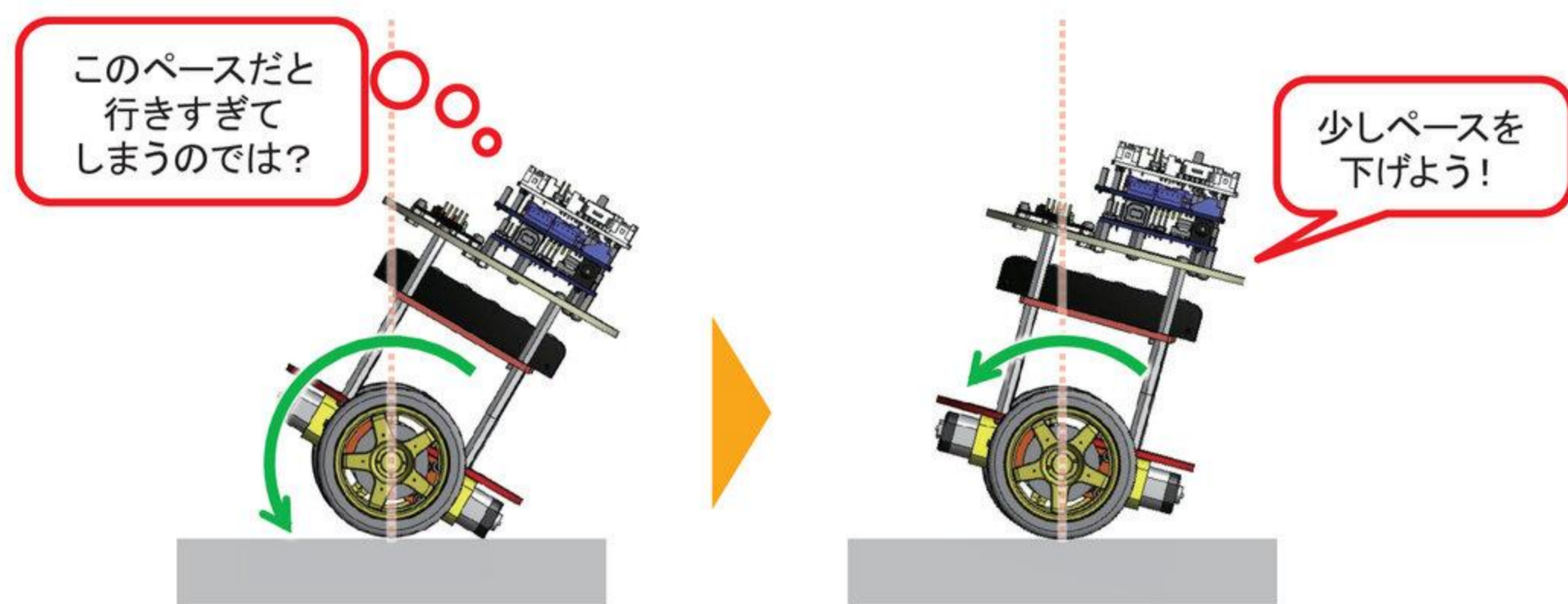


図 1-6 倒立振り子ロボットのオーバーシュートを抑える

「ある一瞬（一点）だけを見たときの、数値変化の大きさ」を求めることを、数学の世界では「微分」といいます。

そのため、このような制御方法を「微分制御」とよびます。微分は英語で「differential」なので「D制御」ともよびます。

こちらは積分制御とはちがい、過去の偏差はいっさい考えません。とにかく今この瞬間の勢いだけを見て、「オーバーシュートしそうかどうか」を予測することになります。

まとめると、比例制御（P制御）に、過去のデータを使う積分制御（I制御）と、未来の予測をたてる微分制御（D制御）をあわせることで、より精密な動きをするロボットを作ろうということになります。

比例制御・積分制御・微分制御をすべて同時に行う場合、とくに「PID制御」と略します。今回完成させる倒立振り子ロボットも、このPID制御を使って姿勢を保っています。

なお、積分制御や微分制御も基本的にゲインが必要です。変数を用いるときは比例ゲイン K_p にならって、それぞれ K_i 、 K_d などとするとよいでしょう。

2. 倒立振り子ロボットを動かす (目安 60 分)

2.0. 倒立振り子ロボットの調整方法

いよいよ倒立振り子ロボットを動かします！ いままで作ってきたどのロボットよりも繊細な調整が必要になります。よく説明を読んで慎重に進めていきましょう。
プログラムでは、「Kp」、「Ki」、「Kd」、そして、「地面に対しての姿勢（姿勢ゼロ点設定）」の値を調整します。

1) 倒立振り子ロボット制御のシミュレート

では、実際に動かしてみたいところですが、調整が難しいため、まずは制御のやり方をシミュレートしてみましょう。

プログラムを実行した後の「倒立振り子ロボット」の状態を図解すると、**図 2-0** のようになります。サンプルプログラム内では、「Kp」と「Ki」の値はすでにある程度調整されています。

そのため、プログラムを実行すると、ロボットが倒れていく傾きを戻そうと矢印の方向に前後にグラグラ揺れて安定しない状態になります。

ですから、まずはこの揺れを抑えるために「Kd」を調整して、安定させていきます。

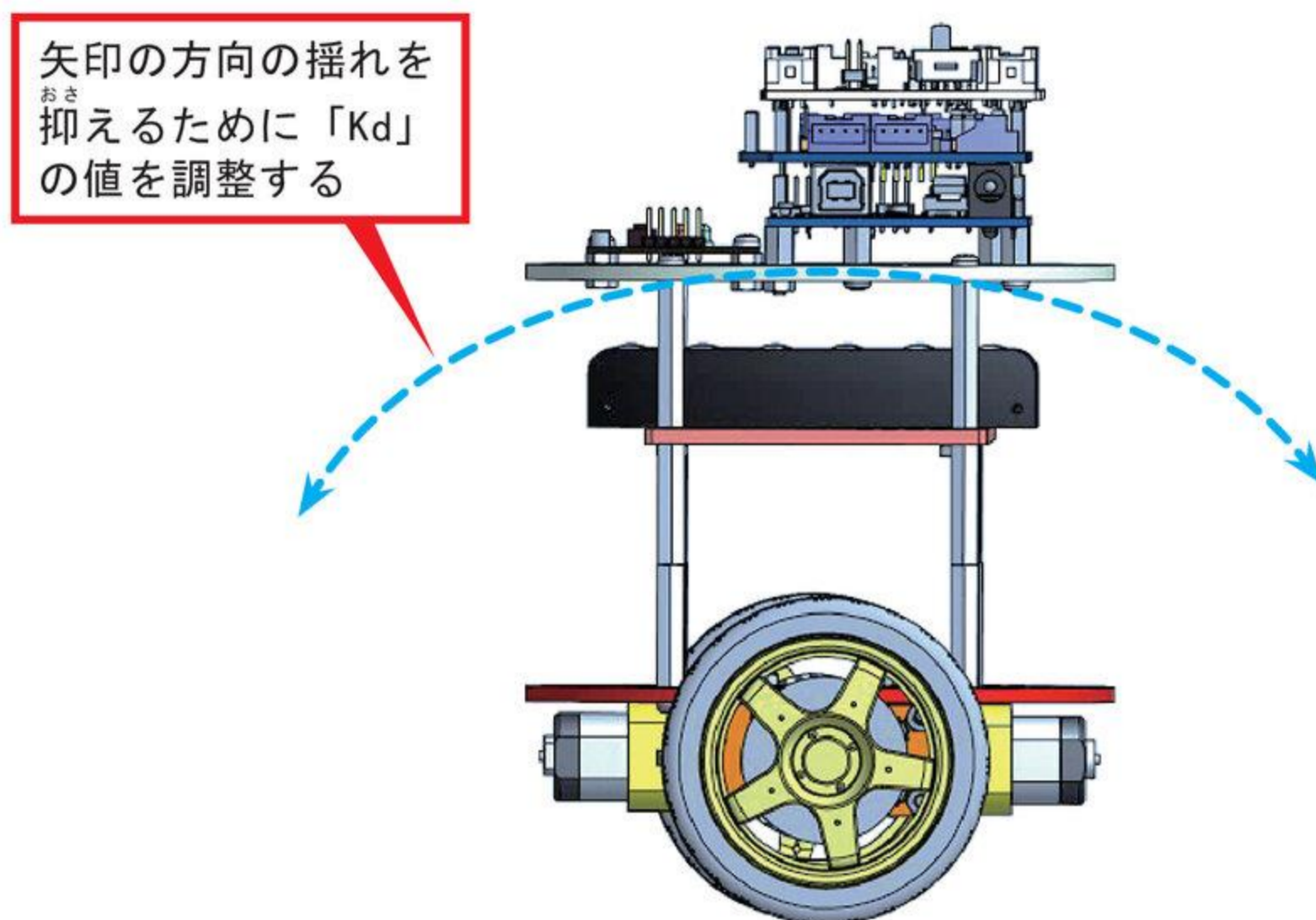


図 2-0 「Kd」の調整

「Kd」を調整していくと、次第にロボットの揺れが少なくなり安定しますが、ロボットの姿勢が地面に対して傾き気味になっていると、意図せずに前進したり、後退したりするので、**図2-1**のように姿勢が地面に対して垂直（ゼロ点）になるように調整します。

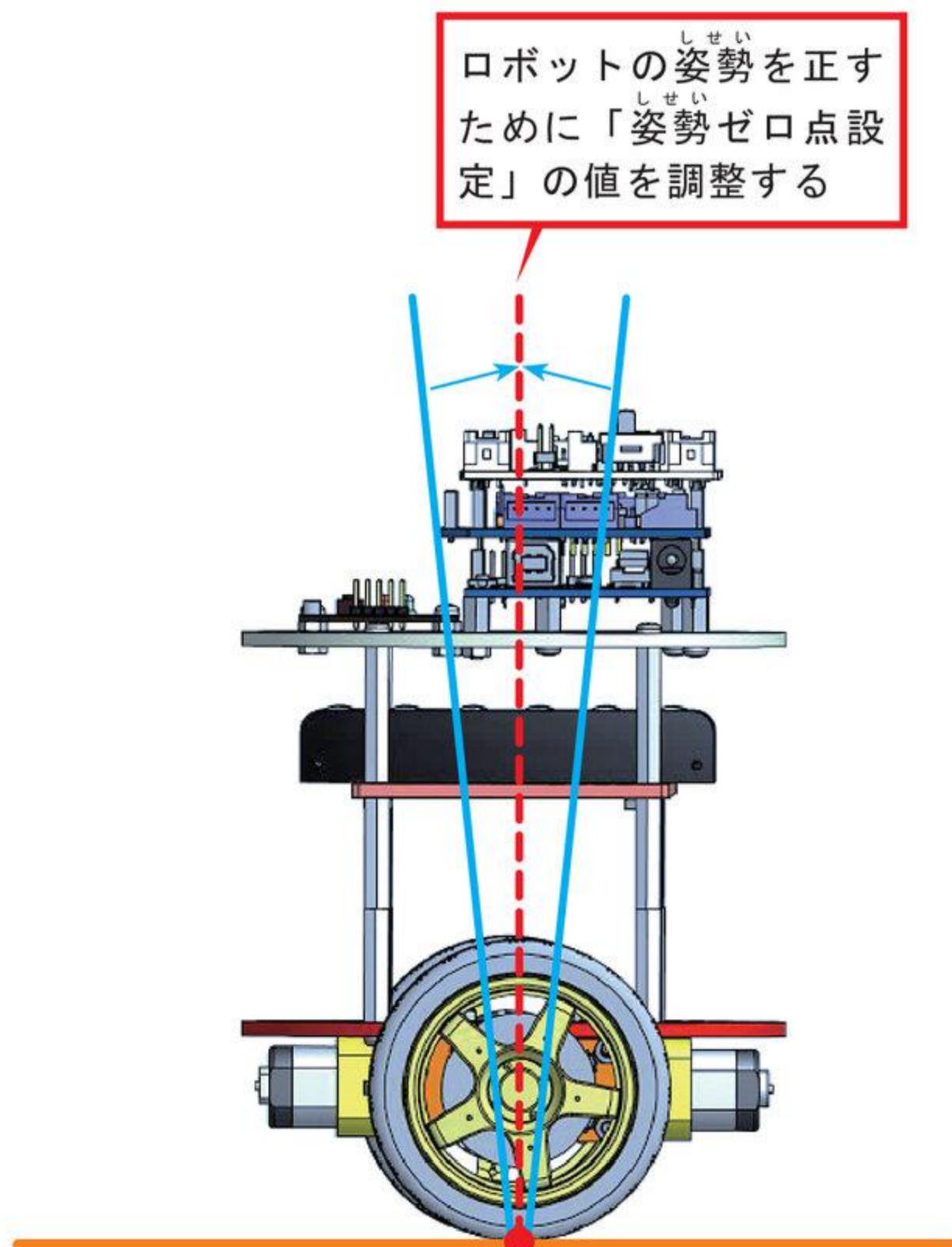


図2-1 姿勢の調整

たとえるなら、人間が片足でバランスを保つときに行う動作と似ていますね。地面から垂直に立ち、からだ揺れないように姿勢を保つのが長く立っていただけるコツです。



図2-2 バランスを保つイメージ図

2) 「Kd」、「姿勢ゼロ点設定」の調整方法と初期位置の調整

では、調整していきましょう。電子回路のボリュームで制御をします。図 2-3 を参照してください。

「Kd」は [A1] で、「姿勢ゼロ点設定」は [A0] でそれぞれ調整します。

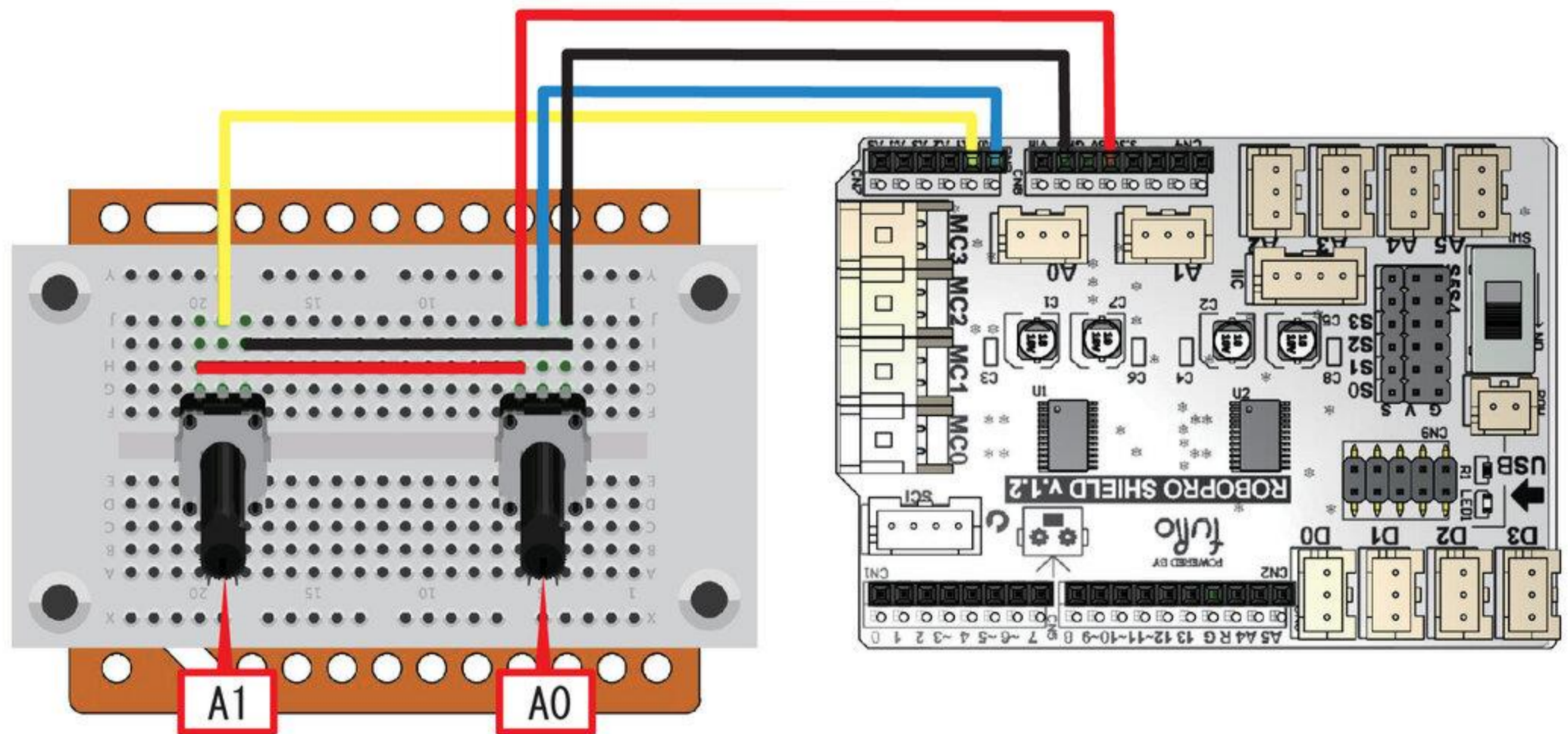


図 2-3 倒立振り子ロボットの電子回路

各ボリュームのつまみの初期位置は、図 2-4 のような向きになります。

[A1] は、つまみのたいらな部分を矢印の方向いっぱいにしておきます。[A0] は、たいらな部分を 180 度の状態にします。



図 2-4 ボリュームの初期位置

3) ^{とうりっしんし}倒立振り子ロボットの状態をシリアルモニターで確認する

続いては、ロボットの調整がしやすくなるように内部の状態を数値で確認できるようにします。数値をシリアルモニターに表示してみます。

^{とうりっしんし}倒立振り子ロボットとパソコンをUSBケーブルでつなぎ、以下のプログラムを実行します。

∞プログラムの書き込み

RoboticsProfessorCourse2 > inverted6 > RemoteSelfBalancing

プログラムを実行すると、ロボットがグラグラと動き出します。

シリアルモニターを開き、通信速度を「115200baud」に設定すると、以下のように「Kp」、「Ki」、「Kd」、「^{しせい}姿勢ゼロ点設定」の4つの値が表示されます。

ボリューム **A1** を初期位置から回していくと、左から3つ目の「Kd」の値が変化します。この値がだいたい1.50くらいになると、多少の揺れこそ残りますが、ロボットは安定してきます。

なお、シリアルモニターを開くときに、ロボットの動作がいったん停止しますが、シリアルモニターに値が表示されたら再び動作します。

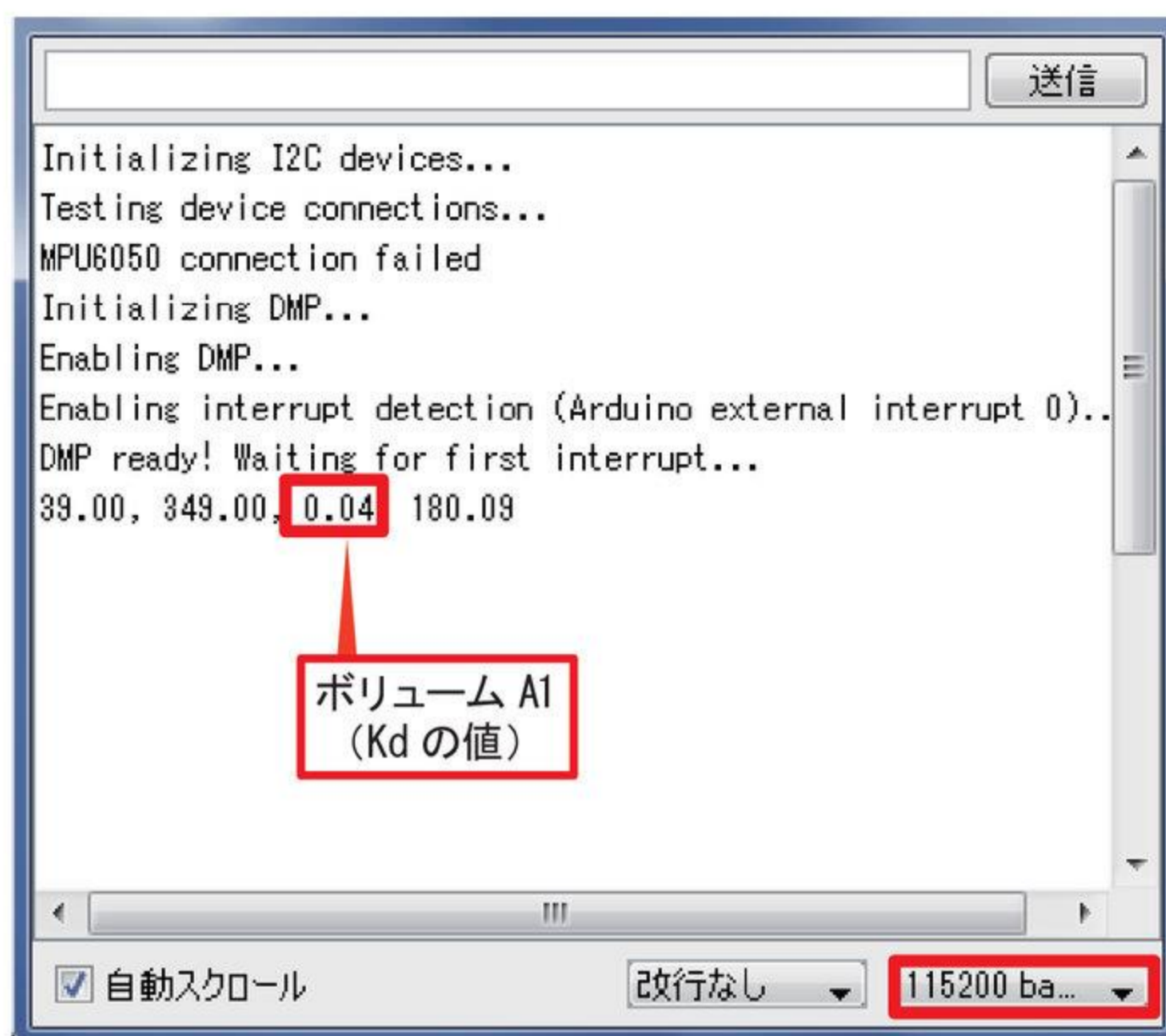


図 2-5 シリアルモニター (通信速度 /115200baud)

ここまで調整ができれば、USBケーブルを抜きましょう。後は実際に動かしながら、調整していきます。

4) 倒立振り子ロボットの動かし方と調整法の確認

では、ロボットを動かす前に、最後に倒立振り子ロボットの動かし方と調整の方法を確認しておきます。

まず、プログラムを実行した後はロボットがグラグラと動きだしますので、安定するまでは必ず手で支えましょう。

停止させたいときは、**図 2-7** のように横に寝かせてマイコンボードのリセットボタンを押しましょう。なお、ロボットを起こすと再び動き始めます。

もしロボットが動かなかったり、モーターが回転し続ける暴走状態になったりしたら、やはりマイコンボードのリセットボタンを押しましょう。

さらに、モーターについているコンデンサーが発熱した場合は、一定時間放熱してから再開してください。

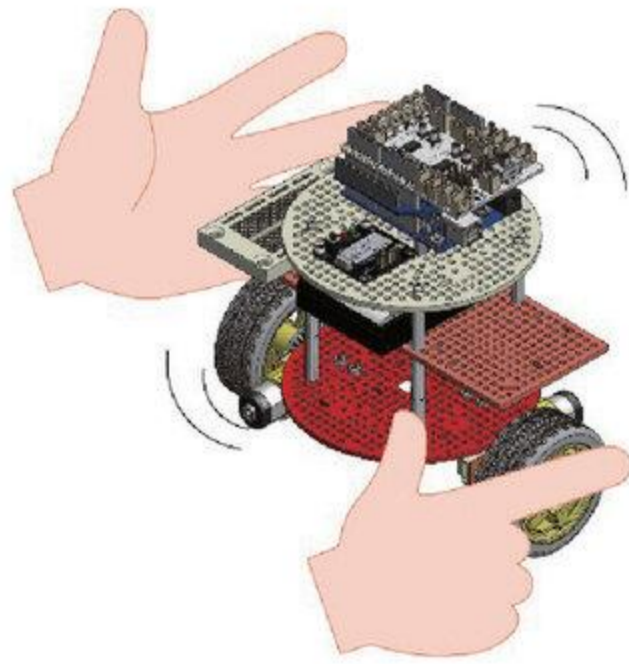


図 2-6 手で支える

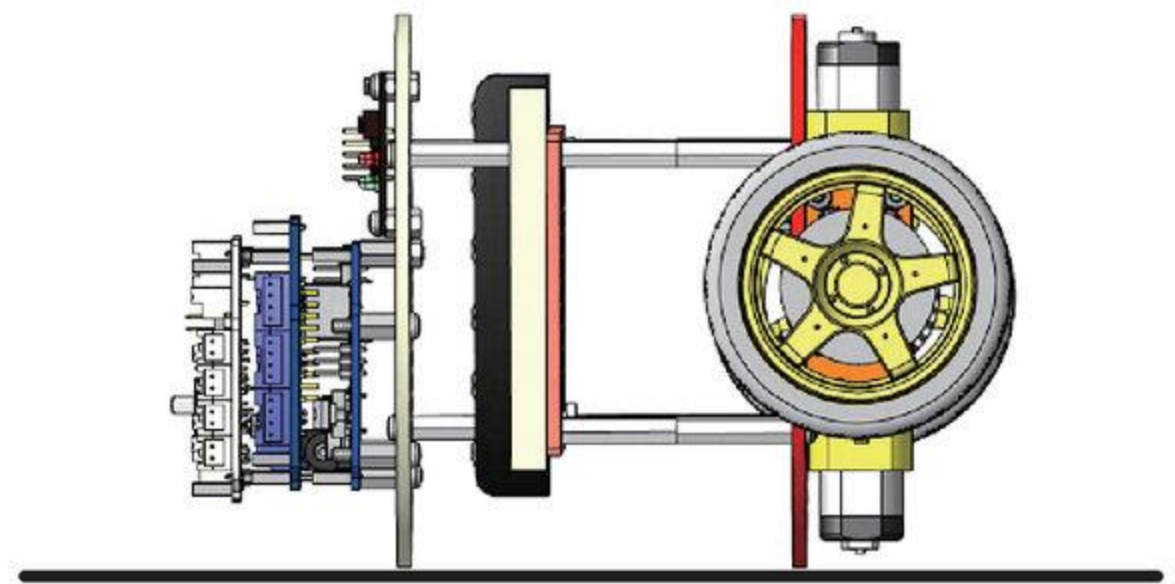


図 2-7 ロボットの停止姿勢

[A1] のボリュームを回すと [Kd] が調整されます。

[A0] を左に回すとオレンジの矢印の方向に、右に回すとグリーンの矢印の方向に、それぞれ角度を補正します。

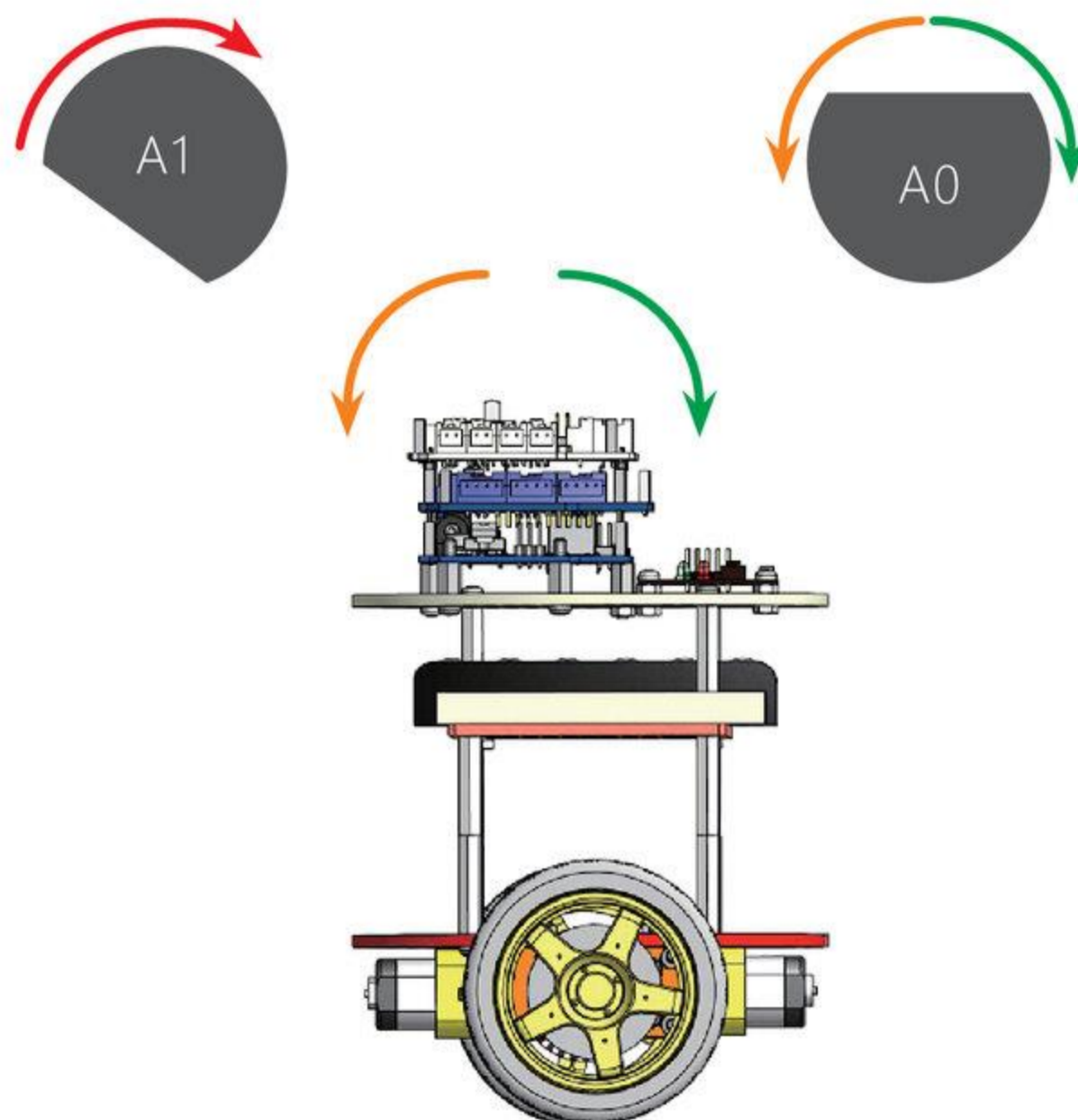


図 2-8 倒立振り子ロボットの調整

ここまでに実際にやってみたものも含まれますが、全体の実行手順をまとめると以下のようになります。

**POINT**

1. USB ケーブルでロボットとパソコンとをつなぐ。
2. プログラムを実行する。
3. シリアルモニターを開く。
4. ボリューム [A1] (Kd) を調整する
5. USB ケーブルを取り外す。
6. ボリューム [A0] (姿勢ゼロ点設定) を調整する。
7. コントローラーをペアリングして制御する。

では、いよいよ動かしましょう。

2.1. 調整しながら動かしてみる

では全体の流れがわかったところで、以下のプログラムを実行し、実際に動かしながら調整してみましょう。

プログラムの書き込み

RoboticsProfessorCourse2 > inverted6 > RemoteSelfBalancing

コントローラーでの操作方法は以下のようになります。

なお、スティックを一気に倒すと、スピードが上がりすぎて倒れてしまうので、ロボットの状態を見ながら徐々に動かしてください。また、左右の旋回は、前後の平行移動をともなわないとロボットが倒れてしまいます。旋回を行う際には、前進または後退と一緒にコントロールしてください。



図 2-9 コントローラー操作

講

コントローラーを使った操作は、床などの安定した場所で行ってください。また、実行する場所によって滑り具合や傾きが異なる場合がありますので、場所を変える場合はその都度調整を行ってください。

いかがでしょうか。うまくロボットを動かすことはできましたか？
すぐにうまくいかなくても、ぜひ少しずつ調整してみてください。

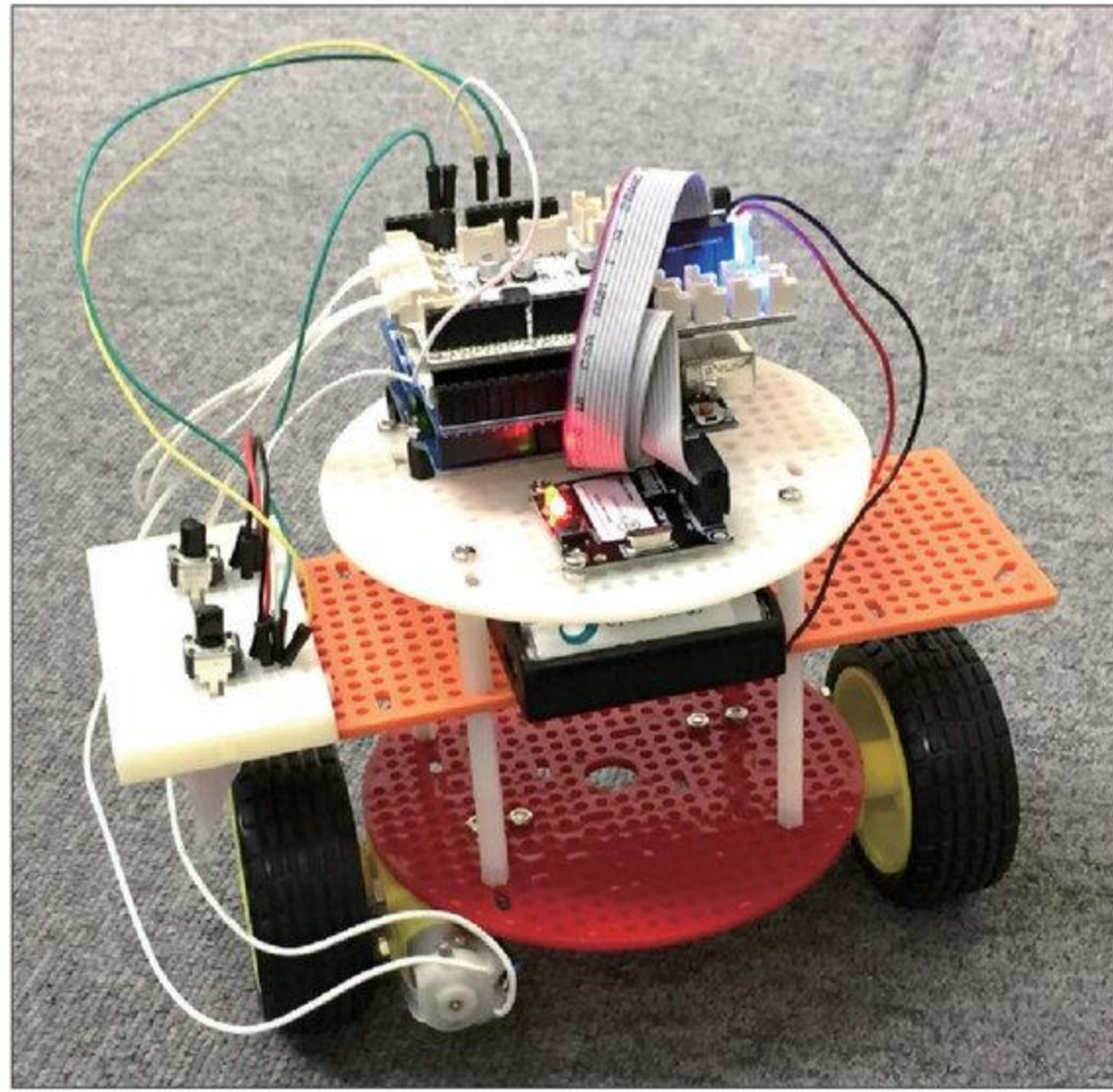


図 2-10 どうりつしんし 倒立振り子ロボットの動作イメージ



POINT

「ロボットがうまく立ってくれない！」というときは、ロボットを持ち上げた状態（タイヤが浮いている状態）で、ゆっくりロボットを前後に傾けてみましょう。傾きに応じてタイヤの回転が変化するようなら、だんだん傾きを減らしていき、タイヤの回転が止まる瞬間を探してみます。タイヤがほぼ止まったな、と思ったら、そのままそっと地面に置き、ゼロ点や Kd の調整を行ってください。

大まかに、立つことは出来るが前か後ろにどんどん進んでしまう（その場にとどまれない）ならゼロ点調整を、その場にとどまってはいるが前後にがたがた振動してしまうなら Kd の調整をするとよい場合が多いです。

そもそも傾き具合に関係なくタイヤの回転が変化しないときは、いったんロボットの電源を切り、数秒待ってからもう一度電源を入れなおしましょう。必要に応じて何度かこの手順をくり返します。

それでも直らなければ、マイコンボードと姿勢検出シールドとの接続部をチェックしましょう。基板をほんの少しだけ（0.5 ミリ程度）浮かせたり、水平にしたり、傾けたりすることで信号のやり取りが変化することがあるので、シールドの位置を微調整してマイコンボードのリセットボタンを押して確認してみましょう。

また、ジャンパー線が断線していたり、マトリクス LED シールドが接続されたままだったりしないかも見ておきましょう。

タイヤの回転は変化するが回転が足りないという場合は、電池残量をチェックしてみましょう。

なお、倒立振り子ロボは直立（まっすぐ立っている）状態を原点として、そこから前後に45度以上傾けると緊急停止するようになっていますが、この原点の位置が直立状態からずれていると、直立させているつもりでもロボットが「緊急停止するほど傾いている」と誤認してしまうことがあります。この時は第3回でも行った、プログラム最終行の値（”+180”の部分）の調整を行いましょう。原点が直立状態から何度くらいずれているかを調べて、その数字だけ値を上下させます。

また、かなり上級者向けの調整になりますが、プログラムのはじめの方に書いてある Kp 、 Ki の値を直接書きかえるという手もあります。

特に、アルカリ乾電池以外の電源（充電式乾電池や AC アダプター）を使う場合、同じ出力値でもモーターの回転具合が大きく変化するので、3つのゲインを変更してしまった方が早いこともあります。

ゲインの目安はプログラム内に書いてあるので、よく読んでみましょう。

講

タイヤをオムニホイールに変えてみても動きます。若干ですが、動き方が変わるので、時間が余ったら試してみてください。

3. まとめ（目安5分）

今回は、PID制御という、倒立振り子ロボットの制御方法を勉強しました。本当は、角度や角速度やゲインの計算などの数学を活用している訳ですが、今回は、基本的な原理が理解できていれば問題ありません。

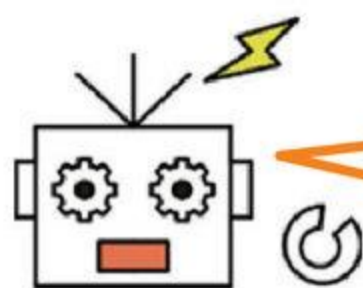
倒立振り子ロボットのパラメータ調整、そしてラジコン化も行いましたが、調整がいかに大変かも体験できたと思います。第6回までに、姿勢検出シールドを使って様々な物理量を検出して体験しました。「目には見えないけれども、確実にあるもの」を検出して、それをフィードバック制御を使ってロボットを動かしました。

ロボットは人間には出来ないほど速く、正確に、たくさんの処理を行ってくれますが、反対に我々人間にとっては「なんとなく」で行える作業でも、ロボットには細々と命令を出さないとならないこともあります。

人間には難しいがロボットには簡単なことがあるように、人間には簡単だがロボットには難しいこともたくさんあるということが、倒立振り子ロボットを通じて理解できたのではないのでしょうか。

これらをロボットにやらせるためには、人間が「なんとなく」でできていた動作をしっかりと分析し、詳しい命令を作りあげていくことが大切です。

日常の動作一つひとつでも、「どんな動きを、どんなタイミングでしているか？」を考えておくと、今後の参考になるかもしれませんね！



次回、また新たなロボットの世界でお会いしましょう！

講

- 以下の授業の目標を再確認します。
 - ・倒立振り子ロボットの制御方法を学ぶ
 - ・倒立振り子ロボットを調整しながら動かす
- 今回のタームで学んだ感想や面白かったことなどを、生徒から聞いてみましょう。
- 次回、2年目コースは「不思議アイテムⅡ」、3年目コースに進級する場合は「不思議アイテムⅢ -1」になります。

《次回必要なもの》

次回は、以下のパーツを持ってきてください。




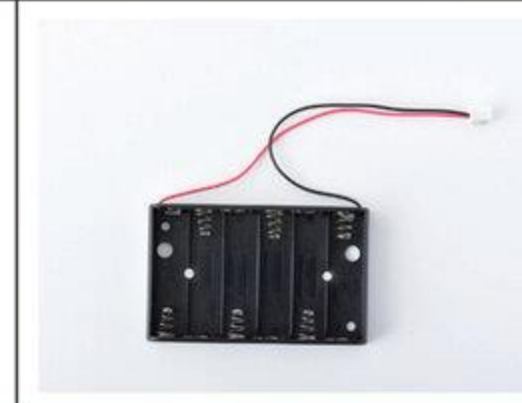



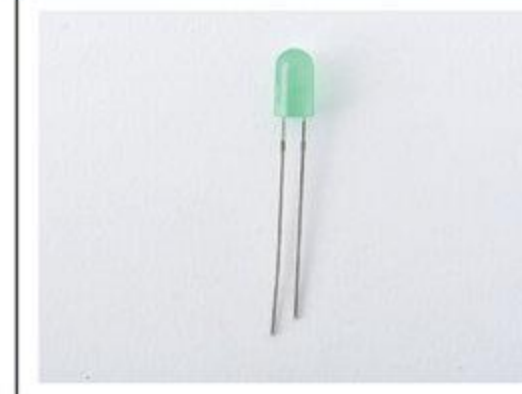
ラジオペンチ 1	USB ケーブル 1	マイコンボード 1	電池ボックス 1
			
301 ブレッドボード 1	ジャンパー線 65	抵抗 (100 Ω / 1k Ω / 10k Ω) 各10	緑色 LED 10
			

図 3-0 次回必要なもの