



ユーザー登録 (/signup?redirect_to=%2FRyosukeH%2Fitems%2Fbf95c58e352774ed8de7)

ログイン (/login?redirect_to=%2FRyosukeH%2Fitems%2Fbf95c58e352774ed8de7)



@RyosukeH (/RyosukeH) 2017年06月18日に更新
(/RyosukeH)



より良い制御を目指して～モデル予測制御の導入～

C++(/tags/C++) Control(/tags/Control) Udacity(/tags/Udacity) MPC(/tags/MPC)
IPOPT(/tags/IPOPT)

👍 23



前回「0からわかるPID制御」

(<http://qiita.com/RyosukeHonda/items/373f6451c4946b1e447e>)という記事を書きました。今回はPID制御の欠点に注目し、それを改良する方法について探りたいと思います。実装は後編で行うことを予定しています。

PID制御の欠点

以前用いた条件設定でPID制御の欠点を考えます。

PID制御の式は以下のように書くことができます。

$$steer = K_p cte + K_D \frac{d}{dt} cte + K_I \sum cte$$

この式を見ると、現在と過去の情報が用いることができていません。そのためコーナーに差し掛かったときは以下のgifのような挙動となります。



(<https://camo.qiitusercontent.com/248a89d9462a5b9af0fb4831acb78298628600b5/68747470733a2f2f71696974612d696d6167652d73746f72652e73332e616d617a6f6e6177732e636f6d2f302f3136333638352f65343761323637632d333563332d346131632d386237662d3461613832393563666530372e676966>)

何が起こっているかというと、

- ・減速せずにコーナーに入る。
- ・CTEが大きくなる。
- ・CTEを小さくするために、急ハンドルを切る。
- ・急ハンドルを切ったために挙動がしばらく振動的になる。

また、別の欠点として、制御量を入力するまでの時間差を考慮することができないことも挙げられます。（0秒で目標のハンドル角に合わせることが不可能）

一方人間が自動車をコントロールする場合は次のようにします。

- ・コーナーを確認する。
- ・コーナーに入る前に減速する。
- ・減速後にハンドルを切り、目標軌道を進む。
- ・加速する。

人間が自動車を運転するように滑らかな軌道を描くためには、どのように制御すればいいのでしょうか？

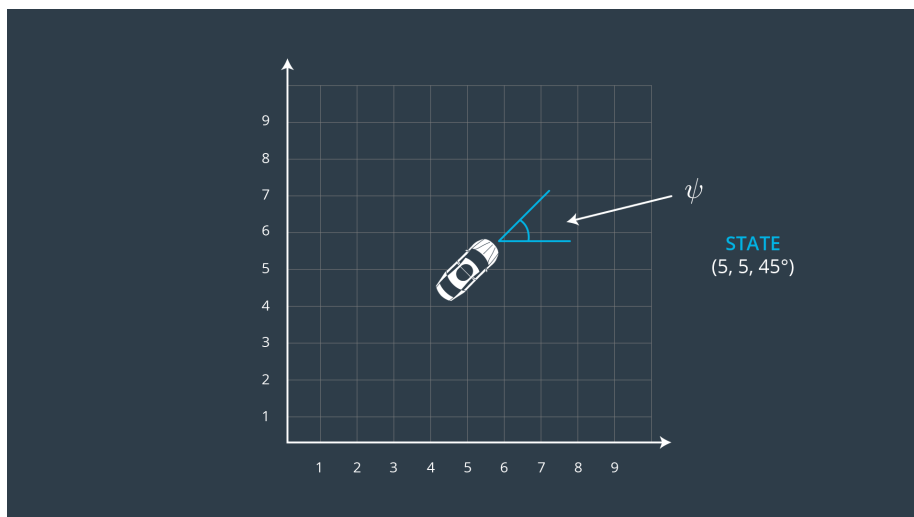
そのような方法の1つに、あらかじめ軌道を予測し、その予測に対して操作量（ハンドル量とアクセル量）を入力するという方法が考えられます。この方法はモデル予測制御を用いることにより、実現することが可能です。今回はこの手法について検討したいと思います。

モデル予測制御とは

モデル予測制御とは「制御対象のモデルを利用し、最適制御問題を解くことにより、将来の予測値を求め、その予測値を用いて制御する方法」です。元々は、サンプリング周期の長い化学プラントや電力プラント等で用いられていましたが、近年の演算処理能力の工場により、サンプリング周期の短い対象についてもリアルタイムに予測することが可能となりました。そこで、今回は自動車シミュレータにモデル予測制御を適応しようと思います。

モデルの構築

モデル予測制御を利用するには、制御対象のモデルを構築する必要があります。そこで、自動車のモデルを構築します。正確なモデリングには、車体に加わる力を詳細に記述したDynamicモデルを用いる必要がありますが、今回は単純なKinematicモデルを用いることにします。



(<https://camo.qiitusercontent.com/b9771c9cad9aa923ebb8edada81f633b931cabff/68747470733a2f2f71696974612d696d6167652d73746f72652e73332e616d617a6f6e6177732e636f6d2f302f3136333638352f35653635343836302d373637372d333961642d623436342d376664332383631366235382e706e67>)

(quoted from Udacity's lecture)

自動車の位置と速度を (x, y, ψ, v) と表し, 入力量を (δ, a) とします. ただし δ はハンドル角, a はアクセル量を表します.

自動車の状態について以下の式が導くことができます.

$$\begin{aligned}x_{t+1} &= x_t + v_t \cos \psi_t dt \\y_{t+1} &= y_t + v_t \sin \psi_t dt \\v_{t+1} &= v_t + a_t dt \\\psi_{t+1} &= \psi_t + \frac{v_t}{L_f} \delta_t dt\end{aligned}$$

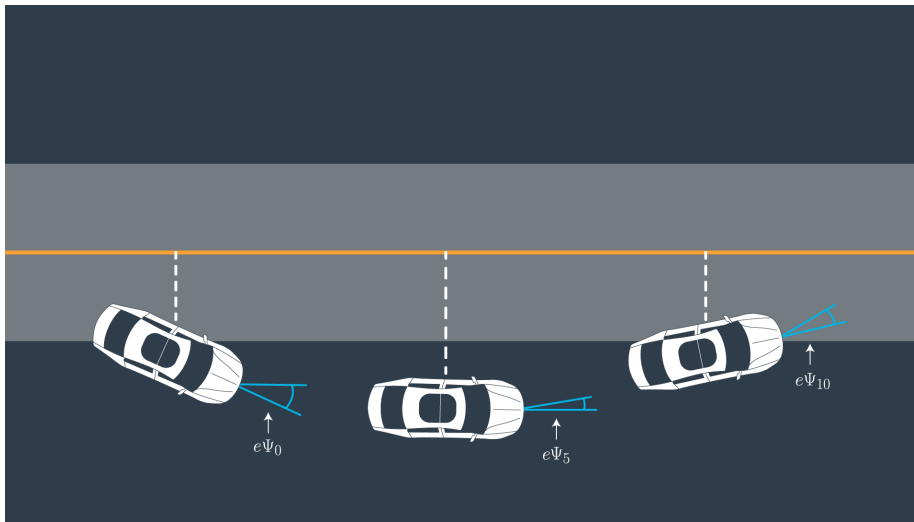
ここで, L_f は自動車の先端から重心までの距離とします.

$\psi_{t+1} = \psi_t + \frac{v_t}{L_f} \delta_t dt$ の式はこちら



(http://www.me.berkeley.edu/%7Efrborrel/pdfpub/IV_KinematicMPC_jason.pdf)を参考にしました.

また, 「0からわかるPID制御」(<http://qiita.com/drafts/373f6451c4946b1e447e/edit>)での問題設定と同じように道路の中心と自動車の位置との距離差CTE(Cross Track Error)を導入します. 今回は角度も導入したので, CTEの基準線と現在の自動車のなす角 $e\psi$ も導入します.



(<https://camo.qiitusercontent.com/291dc463bff378722036e7964d22896c8cda1ef7/68747470733a2f2f71696974612d696d6167652d73746f72652e73332e616d617a6f6e6177732e636f6d2f302f3136333638352f63356138303131322d616264322d376164662d613866332d3038646435303561303133362e706e67>)
(quoted from Udacity's lecture)

$$\begin{aligned}cte_{t+1} &= cte_t + v_t \sin e\psi_t dt \\e\psi_{t+1} &= e\psi_t + \frac{v_t}{L_f} \delta_t dt\end{aligned}$$

と記述することができます.

最適制御問題とは

制御系の設計には, 与えられた目標状態に達することだけが要求されることは少なく, その目標状態に達するまでの途中経過も設計仕様に与えられる場合があります. 今回のシミュレーターの例で言えば, 「人間が自動車を運転しているような挙動を実現するために急ハンドル, 急アクセル等を避けつつ, CTEを小さくすること」です. 「人間が自動車を運転しているような」という条件の達成度をコスト関数で評価し, そのコスト関数を最小にする制御系を設計する問題のことを最適制御問題と言います.

最適制御問題を解くためには、コスト関数を設定する必要があります。そこでコスト関数として、必要なものとしては、以下のものが考えられます。

- ・CTE
- ・CTEの基準線と現在の車のなす角の角度差

以上のコスト関数のみの場合には、目標軌道に達したときには、コストが0となるため、自動車は移動する必要がなくなるため、自動車が静止してしまいます。

そのため、目標速度を決めて、コスト関数に含めるようにします。

また、人間が運転する軌道を再現しようとする場合には、急激なハンドル、アクセル量変化は望まれません。そのため、ハンドル、アクセル量の変化についてもコスト関数に含めるようにします。

すると以下のようなコスト関数ができます。

$$\int_0^T (a \times cte_t^2 + b \times e\psi_t^2 + c \times (v_t - v_{ref})^2 + d \times (\delta_t - \delta_{t-1})^2 + e \times \delta_t^2 + f \times (a_t - a_{t-1})^2) dt$$

(ものすごい式になってしまいました...)

ここで、 v_{ref} は目標速度、 a, b, c, d, e, f, g は重みを表しており、自由に設定することが可能です。

例えば、CTEの重みに大きな値を入れるとCTEを素早く減少させるような操作入力 that 得られます。

また、それぞれのコストの単位が異なるため、重みは慎重に決定する必要があります。

この最適制御問題を解くためにはハミルトン・ヤコビ方程式と呼ばれる偏微分方程式を解く必要がありますが、理論が複雑なため割愛します。実装する際には、ある制限に対して偏微分方程式を解いてくれるライブラリーが存在しますので、それを用いて実装したいと思います。IPOPT(Interior Point OPTimizer)というライブラリーを使用します。

実装については次回行う予定です。

ちなみにモデル予測制御の結果は下のgifようになります。



(<https://camo.qiitusercontent.com/84e9340f2d10ff90acce88cb8abf1e6a71f3949/68747470733a2f2f71696974612d696d6167652d73746f72652e73332e616d617a6f6e6177732e636f6d2f302f3136333638352f30653634396330382d653363392d376439642d313937342d3937663433363266636364372e6769666>)

黄線が基準線で緑線が予測値となっています。

参考文献

最適制御問題については以下の本を参考にしました、

現代制御論 (https://www.amazon.co.jp/gp/product/4339032123/ref=as_li_tl?ie=UTF8&camp=247&creative=1211&creativeASIN=4339032123&linkCode=as2&tag=ryo519-22&linkId=f30d36b042714fc9c0f5132ffcc26bab)

(<https://camo.qiitusercontent.com/c648d945170cea252fb8e9da7ee620fcab05e5ad/68>)

747470733a2f2f69722d6a702e616d617a6f6e2d616473797374656d2e636f6d2f652f69
723f743d72796f3531392d3232266c3d616d32266f3d3926613d343333393033323132
33)

📁 ストック

👍 いいね

🗨️ 23

(<https://qiita.com/RyosukeH/items/bf95c58e352774ed8de7/likers>)

Tweet

1

(<https://qiitadon.com/share?text=%F3%R2%RR%F3%R2%RA%FR%RQ%AF%F3%R1%R4%F5%RR%R6%F5%RF%A1%F3%R2%Q2%F7%QR%AF%F6%RC%R7%F3%R1%Q7%F3%R1%A6%F3%R0%Q7%F3%R3%A2%F3>)

Like 1

9




@RyosukeH (/RyosukeH)


自動運転, AIに興味があります.


(/RyosukeH)


👤 フォロー



- 関連記事 Recommended by (<http://www.logly.co.jp/privacy.html>)
- 


0からわかるPID制御 (実装編) (<https://qiita.com/RyosukeH/items/373f6451c4946b1e447e>) by RyosukeH
- 

0からわかるPID制御 (<https://qiita.com/RyosukeH/items/9e5ce2ebdadd90e3db00>) by RyosukeH
- 

WSDM 2016 勉強会: Feedback Control of Real... (https://qiita.com/komiya_atsushi/items/f34fcfaa8215321b1fb9) by komiya_atsushi
- 

OpenGoddard使い方1-非線形最適制御・軌道生成のためのpythonライ... (<https://qiita.com/ina111/items/ec73bc77ad45dd65ac1e>) by ina111

🔗 この記事は以下の記事からリンクされています



より良い制御を目指して～モデル予測制御 (実装編)～ (/RyosukeH/items/b947efa9dbcefa396baa#_reference-945595abb2eafaa195bc) からリンク 11ヶ月前

あなたもコメントしてみませんか :)

ユーザー登録(無料) (/signup?redirect_to=%2FRyosukeH%2Fitems%2Fbf95c58e352774ed8de7%23comments)

すでにアカウントを持っている方はログイン (/login?redirect_to=%2FRyosukeH%2Fitems%2Fbf95c58e352774ed8de7%23comments)