

乗り心地を重視した車両制御システムの設計とシミュレーション Designing and Simulation of Vehicle Control System in Consideration of Ride Quality

服部 有里子[†] 青木 慎吾[†] 小林 颯[†]
Yuriko Hattori[†] Shingo Aoki[†] Hayate Kobayashi[†]

1. はじめに

交通事故死者削減・渋滞低減のため、自動車の自動走行・高度運転支援技術の開発が進められている。車の機能は、走る・止まる・曲がるに加えて乗り心地を考慮する必要がある。本稿では自動車の車両速度制御について自動運転制御により、乗り心地を重視した車両制御システムを開発した。

「乗り心地を重視した車両制御システム」では、走行中に車両速度を追従制御し、ドライバがアクセルペダルを操作しなくても自動的に車両速度を一定に維持できるクルーズコントロールシステムを構築する。クルーズコントロールでは、車両速度を動的に計算し、最適なペダル開度の設定による、安定した乗り心地のよい運転制御システムの開発が重要である。この場合、要求されるシステムの課題は以下の2点である。

- 車両の速度変化を動的に計算し、車両速度を一定に維持することで、実車速が急変化することなく、走行安全性を優先すること。
- 外乱がある場合もクルーズコントロールすることにより、振動を最小限に抑え、即時に車両速度を追従制御すること。

これまで車両制御システムとして、フィードバック制御系のPID制御(Proportional, Integral and Differential制御: 制御偏差に基づいて制御入力信号を生成する制御)のパラメータ調整方法が提案されている[1][2]。しかし、実際の車両モデルを考慮した場合と比較すると、たとえ正確にモデリングできたとしてもモデル誤差が少なからず存在する。本稿では、モデル誤差についてPID制御のパラメータ調整法を適用し、制御対象である自動車の乗り心地を重視することにより、車両制御の安定化を図った制御システムの設計とシミュレーションについて報告する。具体的には、車の自動速度制御を行う、フィードバック制御系としてPID制御を設計した。シミュレーションにより検証・評価を行い、自動走行運転の速度制御による乗り心地向上、及び快適性向上について、サービスの有効性を評価した。

以下、2章で関連研究、3章で問題設定と要求条件について述べたのち、4章で乗り心地を重視した車両制御システムを提案し、5章で具体的なフィードバック制御系のPID制御の設計について解説する。6章でシミュレーションによる検証・評価について説明し、最後に7章で本稿をまとめる。

2. 関連研究

車両制御システムとしては、フィードバック制御系のPID制御のパラメータ調整方法が提案されている。しかし、実際の車両モデルを考慮した場合と比較すると、モデル

誤差が少なからず存在する。また、車両重量、外乱による環境の変化による各パラメータの変動等を考慮すると、モデルの安定性は重要な要素であると考えられる。機械系においてPID制御は広く利用されているが、対象が時変系であるためにPIDパラメータの調整がうまくできない場合が往々にして存在している。本稿では、そのような問題に対してアドバンス制御であるセルフチューニング法やロバスト制御の観点からPIDパラメータを調整する方法について研究した。

3. 問題設定と要求条件

3.1 PID制御

本研究では、フィードバック制御系として機械系で広く利用されているPID制御を採用する。PID制御は入力値の制御を出力値と目標値との偏差、その積分及び微分の3つの要素によって行う方法である[3]。

3.2 パラメータ調整

PID制御の各パラメータの値(Kp, Ki, Kd)は設計値であるため、制御対象と独立に自由に与えることができ、要求を満たす最適値を決定する必要がある。しかし、一般的な制御対象に対して、最適なゲイン値を与える代表的な方法は存在せず、実際の調整は数値シミュレーションを行ったり、実際の対象物に調整器を繋いで試行錯誤的に最適値を得ている。

パラメータ調整の経験則として、ジョン・ジエグラ(John G. Ziegler)とナサニエル・ニコルス(Nathaniel B. Nichols)らにより提案された2つの調整法、ジエグラ・ニコルスの限界感度法とジエグラ・ニコルスの過渡応答法が知られている[4]。

3.3 要求条件

本稿で提案する乗り心地を重視した車両制御システムでは、これらの課題と特性に対応した安定性の高いシステムが要求される。ここで、システムが満たすべき要求条件を以下にまとめる。

- ① アクセルペダルを踏まなくても一定の速度を保持して走行するシステムであること。
- ② 横風などの外乱を受けた場合も外乱による操舵角や位置のずれを算出し、走行方向を維持するシステムであること。
- ③ ブレーキを踏み込むと、クルーズコントロールが解除されること。

4. 乗り心地を重視した車両制御システムの提案

前章で整理した要求条件を満足するシステムとして、乗り心地を重視した車両制御システムを提案する。車両

[†] 国立大学法人筑波技術大学, Tsukuba University of Technology

制御システムは、エンジン・トルクコンバータ (T/C) ・車両側・変速制御の4つのサブシステムから構成される。

エンジン回転数は、T/Cにて入力トルクと出力トルクとして伝達される釣り合いトルクの差を使用して余剰トルクとして計算し、エンジン回転数の回転上昇に使用される。T/Cにて計算した出力トルクから回転角加速度を計算し、回転角速度から速度に変換し、それを変速制御に送り込む。車両速度に応じて、空気抵抗を計算する。変速制御は、適切な時期にどのギア段を使うかを判断する装置である。乗り心地を重視した車両制御システムのシステム構成を図1に示す。

5. フィードバック制御系のPID制御の設計

5.1 エンジン

エンジントルク[Nm]は、シミュレーションスタート後はデータマップと無関係に0.2秒間、100[Nm]のトルクを発生する。0.2秒経過した後はデータから計算した値に切り替える。

ペダル開度[%]と現在のエンジン回転数[rpm]から入力されたデータによりエンジントルク[Nm]が出力される。

5.2 トルクコンバータ

5.2.1 速度比 e の計算

エンジン回転数 N_e [rpm]、出力回転数 N_{out} [rpm]とし、以下の数式を満たす。

駆動側速度比: $e = N_{out}/N_e$ [0~1]

被駆動側速度比: $e = N_e/N_{out}$ [0~1]

ただし、 $N_{out}=0$ のとき、 e は 0 とする。

容量係数: $C = f_1(e)$ [Nm/rpm²]

トルク増幅比: $T = f_2(e)$ [Nm]

釣り合いトルク駆動側: $T_{out1} = C \times N_e^2$ [Nm]

釣り合いトルク被駆動側: $T_{out1} = -C \times N_{out}^2$ [Nm]

出力トルク: $T_{out2} = T_{out1} \times T$ [Nm]

5.2.2 トルクコンバータの全体モデル

釣り合いトルク駆動側: $T_{out1} = C \times N_e^2$ [Nm]

釣り合いトルク被駆動側: $T_{out1} = -C \times N_{out}^2$ [Nm]

のように、正負の符号を切り替える手法を加えて釣り合いトルク・出力トルク・速度比の全体の計算を整理する。

エンジン回転数 N_e は、 T_{out1} と T/C に入力されるトルクを T_e とすると、釣り合いトルク ($C \times N_e^2$) と T_e の差を使用して計算する。 N_e は次の式で計算する。

$$N_e = \int \frac{T_e - T_{out1}}{I_{ne}} ds$$

I_{ne} : 入力イナーシャ[kgm²] (I_{ne} を 0.2[kgm²] とする。)

入力したトルク T_e は、釣り合いトルク ($C \times N_e^2$) は出力トルクとして伝達され、余った余剰トルクはエンジン回転数 N_e の回転上昇に使用される。

5.3 車両側

5.3.1 速度計算

入力トルク[Nm]、ギア段、車両質量[kg]、タイヤ半径[m]、勾配[rad]といった5つの入力から入力軸[rpm]と速度[km/h]を算出する。

回転数 n [rpm] とタイヤ半径 r [m] から速度 V [km] を計算する。

$$V = n \times (2 \times \pi \times r) \times \frac{1}{1000} \times 60 \text{ [km/h]}$$

5.3.2 走行抵抗 (勾配・道路抵抗) の計算

勾配抵抗 R_2 [N] の式:

$$R_2 = m \times g \times \sin \theta$$

道路抵抗 R_3 [N] の式:

$$R_3 = m \times g \times \cos \theta \times C_{rr}$$

m : 車両重量[kg] g : 重力加速度[m/s²] θ : 勾配[rad]

C_{rr} : 転がり抵抗係数

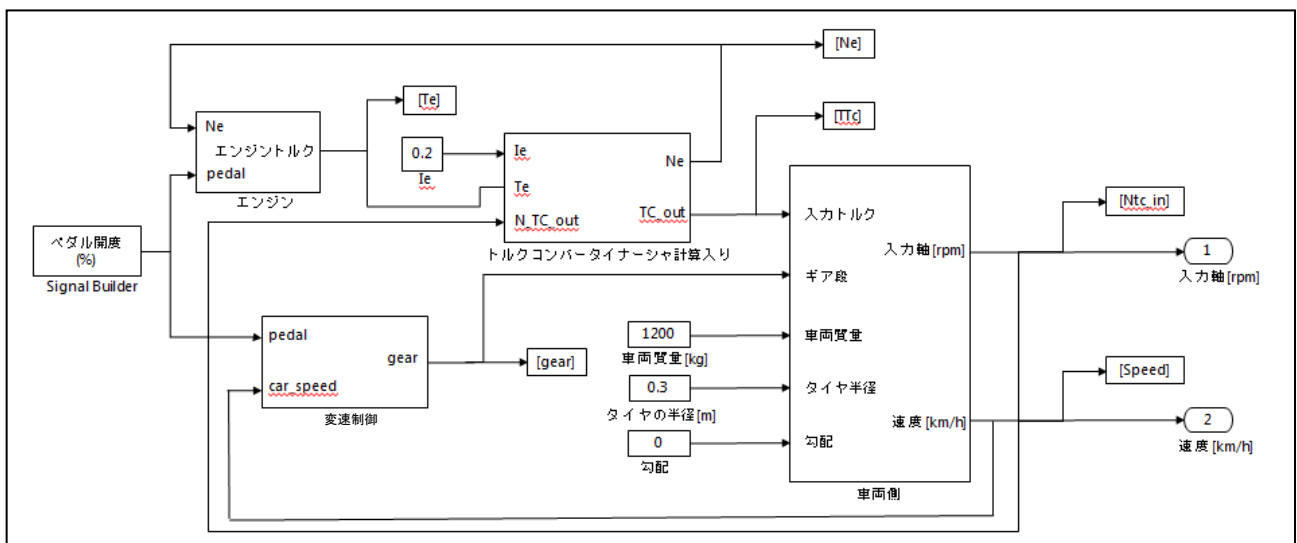


図1 乗り心地を重視した車両制御システムの構成

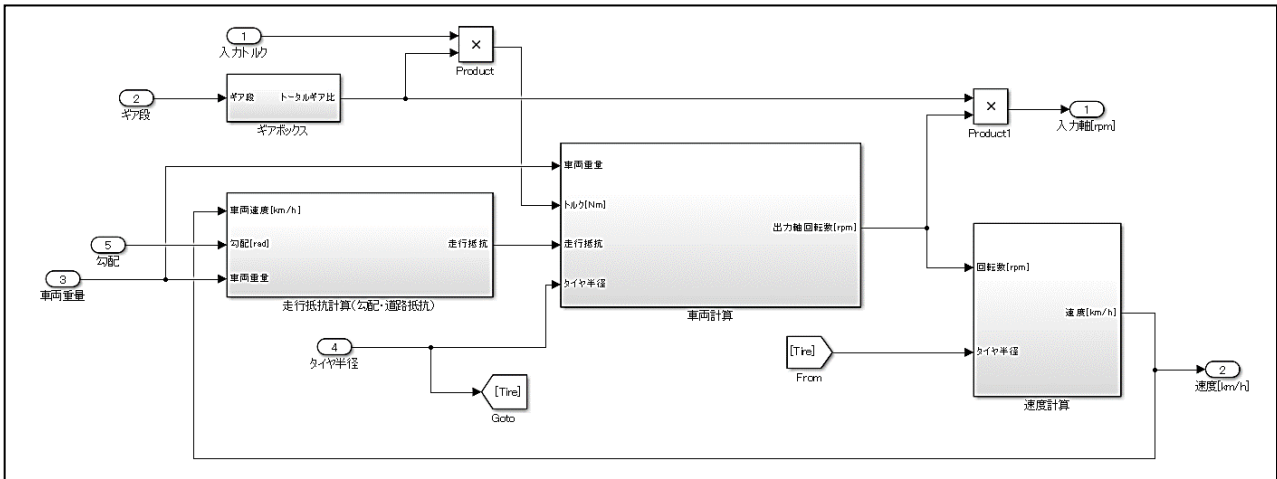


図 2 車両側モデル

5.3.3 出力軸回転数の計算

車両重量 m [kg], タイヤ半径 r [m], 走行抵抗 R [N], トルク T [Nm] から出力軸回転数 n [rpm] を計算する。

車両重量 m [kg], タイヤ半径 r [m] から回転慣性モーメントであるイナーシャを求め、イナーシャとトルクから出力回転数を割り出すことができる。

イナーシャ I [$\text{kg} \cdot \text{m}^2$] の式は次のとおりである。

$$I = m \times r^2 \quad m: \text{車両重量}[\text{kg}] \quad r: \text{タイヤ半径}[\text{m}^2]$$

走行抵抗によって、外からタイヤにトルクがかかり、車両からタイヤに入るトルクから損失トルクが発生する。よって、損失トルクを考慮してタイヤにかかる正味のトルク（損失込みトルク）を計算する必要がある。

損失込みトルク T_{out} [N] の式は、

$$T_{out} = T - T_r \quad T: \text{タイヤに入るトルク}[\text{Nm}] \\ T_r: \text{損失トルク}[\text{Nm}]$$

次にイナーシャ I [$\text{kg} \cdot \text{m}^2$] と損失込みトルク T_{out} [Nm] から回転角加速度 a [rad/s^2] を求める。

回転角加速度 a [rad/sec^2] を求める式は、

$$a = \frac{T_{out}}{I} \quad [\text{rad}/\text{sec}^2]$$

回転角加速度 a [rad/sec^2] を積分して、角速度 n [rad/sec] を求める。式は次のとおりである。

$$n = \int a \, ds \times \frac{1}{2\pi} \times 60 \quad [\text{rpm}]$$

5.3.4 車両側モデル

これまで計算したモデルを組み合わせることで、図 2 に示す車両側モデルが構築できる。

5.4 変速制御

変速制御は、速度などに応じてシフトアップ・シフトダウンするかを判断する制御である。

ペダルを踏むほど速度が上がるとき、shift1-2UP 線を通過するとギア段が 1 から 2 へと変わる。同様に shift2-3UP 線を通過するとギア段が 2 から 3 へと変わる。shift3-4UP 線を通過するとギア段が 3 から 4 へと変わる。

ペダルを緩めると速度が下がってくるため、shift4-3DOWN 線を通過するとギア段が 4 から 3 へと変わるという構造である。

6. シミュレーションによる検証・評価

6.1 フィードバック制御系の検証・評価

6.1.1 P・PI・PID 制御の評価

フィードバック制御系を用いてペダル開度[%]を制御し、下記の要求仕様を満たすパラメータを設定する。

要求仕様：20 秒後に 60[km/h] の速度に到達すること。

P 制御・PI 制御・PID 制御モデルを作成する。パラメータ値は、 $K_p=4$, $K_i=1$, $K_d=0.1$ とする。また、外乱の影響はないものとして外乱の計算は省く。作成した P・PI・PID 制御モデルをそれぞれ車両モデルに取り付けてシミュレーションを実行した [5]。目標値の設定として、ステップ時間を 20[s]、初期値を 60[km/h]、最終値を 60[km/h] とした。3 つの制御系を使用してペダル開度[%]と速度[km/h]を調べた結果、PI 制御と PID 制御による結果はほとんど同じになっていることがわかった。P 制御については初期の加速する部分は緩やかになっているが、要求仕様である「20 秒後に 60[km/h] の速度に到達すること」に達していないことがわかる。P 制御の速度グラフを見てみると、58[km/h]あたりにとどまっていることから、P 制御では目標速度に収束することが難しいと考える。一方、PI 制御と PID 制御では 20 秒後丁度に目標設定である 60[km/h] に収束していることがわかる。ただし、初期の加速する部分は急になっており、5 秒以内に 60[km/h] に達しているため、乗り心地が良くないだろうと考える。

そこで、加速する部分のグラフが緩やかになるように、表 1 に示すパラメータ値を入れ替えてシミュレーションを

実行した。今回は PI 制御と PID 制御の結果がほとんど同じであるため、パラメータが 1 つ少ない PI 制御を使用した。

表 1 PI 制御のパラメータ値

No.	Kp	Ki
1	6	1
2	2	1
3	4	10
4	4	0.1

6.1.2 PI 制御の評価

PI 制御において、表 1 のパラメータ値を用いてシミュレーションを実行した。まず、 $K_p = 6$ 、 $K_i = 1$ と $K_p = 2$ 、 $K_i = 1$ のように元のパラメータ値 $K_p = 4$ 、 $K_i = 1$ において K_p だけ数値を変えて実行した結果、 K_p の数値が大きいほど急速に目標値に近づくことがわかった。逆に、小さいほど緩やかになっている。ただし、この値が小さいほど目標値を大きく超えてしまい、さらに元に戻そうとしても目標値を行き過ぎてしまい、結果的に振動が起きてしまうことがわかった。また、 $K_p = 4$ 、 $K_i = 10$ と $K_p = 4$ 、 $K_i = 0.1$ のように K_i の数値だけ変えて実行した結果、 K_i の数値が大きいほどグラフの波形の振幅が大きくなることわかる。つまり、 K_i の数値が小さいほど収束が早くなるということがいえる。

以上の 4 つのパラメータ値から、加速が緩やかかつ収束が早いグラフにするためには、 K_p と K_i の数値を両方とも小さくする必要があると考える。ただし、どちらかが一方的に小さくなりすぎるとグラフが乱れてしまうため、2 つのパラメータ値を用いてうまく調整する必要がある。4 つのパラメータ値を基にいろいろな数値を入れて試行錯誤を重ねた結果、 $K_p = 2$ 、 $K_i = 0.1$ が最も適した数値であると推測した。その結果を図 3 に示す。

安定走行するまでに 20[sec]かかるが、オーバーシュートや振動がないことが確認できる。実際のシステムで過度な速度変化は車両の損傷や安全性や乗り心地に関わる点であるため、その観点からこの結果は良いといえる。

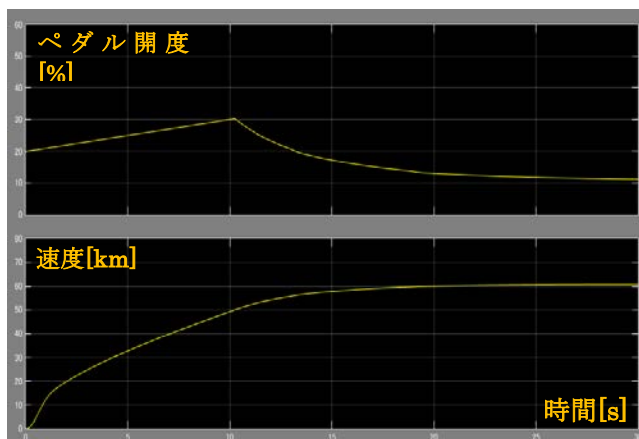


図 3 PI 制御の結果 ($K_p=2$, $K_i=0.1$ のとき)

6.1.3 パラメータ値を変えずに加速度を調整する方法

次に、パラメータ値を変えずに加速度を調整する別の方法を考えてみた。Saturation ブロックを使用して入力信号を上限と下限の飽和値に制約する方法である。これまでの実験では Saturation ブロックにおける上限は 10、下限は -10 である。今回の実験では、上限を 5、下限を -5 に変えて PID 制御のフィードバック制御系を使用する。その結果を図 4 に示す。上限 10、下限 -10 の場合よりも加速度が緩やかになってきたことがわかる。その理由は、上限 10、下限 -10 の場合と比べて上限 5、下限 -5 の方がフィードバックの回数が多くなり、目標値に近づくのに時間がかかるため、加速度が緩やかになっていると考える。

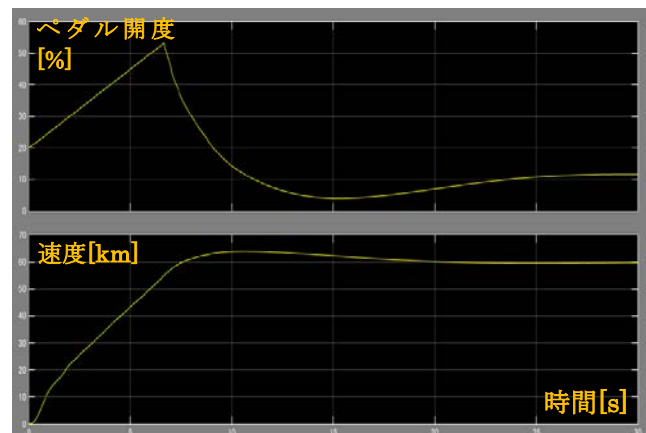


図 4 PID 制御の結果 (上限 5、下限 -5 のとき)

6.2 PI 制御と PID 制御の比較評価

これまでの実験で PI 制御と PID 制御におけるシミュレーション結果がほとんど同じだった。PI 制御における結果を見ると、速度 [km] の波長の振動があまり見られず、収束も早くなっていることから、PID 制御に変えても些細な変化に気づいていない可能性があるということである。そこで、意図的に PI 制御のパラメータ K_p 、 K_i の数値を変えて波長の振幅を大きくさせ、その状態で PID 制御に変えて波長の変化を調べる実験を行った。 K_i の数値が大きくなるほど振幅が大きくなるため、 $K_p=1$ 、 $K_i=5$ 、 $K_d=0.1$ とした。シミュレーション実行時間は 300[s] とした。図 5 に PI 制御の結果、図 6 に PID 制御の結果を示す。

図 5 と図 6 を比較すると、150[s]あたりのところで振幅の大きさがわずかながら図 6 の PID 制御の方が小さいことがわかった。波長の振幅の大きさが小さいほど収束は早くなるため、PID 制御の微分動作 (D 動作) は波長の振動を抑制する性質を持っているといえる。

次に、 K_d の数値を大きくしてシミュレーションを実行した。 $K_d=3$ とした結果を図 7 に示す。 K_d の数値が大きいほど、波長の振幅の大きさが小さくなってきており、収束も早くなっていた。

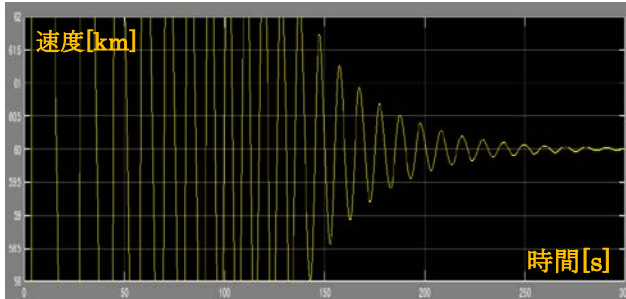


図5 PI制御の結果 (Kp=1, Ki=5 のとき)

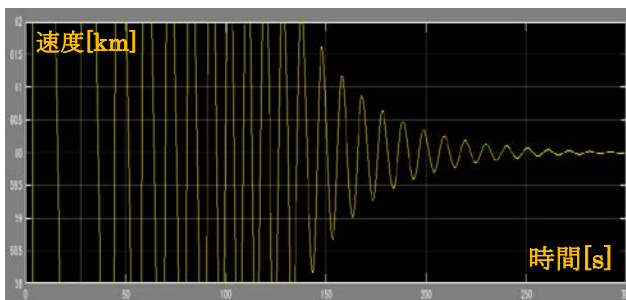


図6 PID制御の結果 (Kp=1, Ki=5, Kd=0.1 のとき)

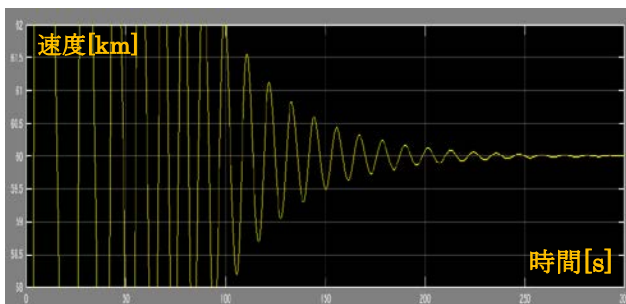


図7 PID制御の結果 (Kp=1, Ki=5, Kd=3 のとき)

PID制御は、目標値と現状の制御量との差（偏差）に比例して操作量を変化させる比例動作（P動作）と、偏差をなくそうとする性質を持つ積分動作（I動作）と、波長の振動を抑える性質を持つ微分動作（D動作）を組み合わせることで、偏差がより少なく収束がより早く安定した応答が実現できると考える。

6.3 外乱がある場合の検証・評価

外乱がある場合、P・PI・PID制御においてどのフィードバック制御系が最も適しているのか検証を行った。外乱として、スタート時点から坂道を走行するものとする。ペダル開度[%]と勾配[°]を設定し、勾配に入っても速度が下がらないようにフィードバック制御を行った。

Kp=4, Ki=1, Kd=0.1とした。3つのシミュレーション結果から、P制御において最初は目標値（60[km/h]）に向かって速度を上げようとしているものの、勾配という外乱が原因なのか30[km/h]にさえも達しておらず、外乱がない状態よりもかなり偏差が残っていることがわかる。

このことから、P制御では外乱がある場合は速度をコントロールすることが難しいと考える。

PI制御とPID制御において、ほとんど同じ結果となった。いずれも50[s]までに目標設定値である60[km/h]に収束しているように見えた。本当に収束しているのかまたどのような波長になっているのか確認するために速度の表示範囲を59[km/h]~61[km/h]に設定した。PI制御とPID制御において違いはほとんど見られなかった。PI制御において波長の振動もほとんど見られず、振動を抑制する役目を果たすD動作を加えても変化が起きないことが原因だと考える。いずれも目標値である60[km/h]に向かって収束しつつあるが300[s]になってもまだ偏差(-0.2~0)が残っていることがわかった。偏差があるため、偏差をコントロールできるI動作の数値をKi=5に変更した。Kiの数値を大きくした結果、より振動的になったが0.03程の偏差は残っているものの、目標値である60[km/h]により近づいた。波長の振動を抑えるために、D動作をKd=1に変えてみた結果、より波長の振幅の大きさが小さくなった。図8にPI制御の結果、図9にPID制御の結果を示す。波形からは、外乱に対してはかなりの強さを持っていると考えられる。



図8 PI制御の結果 (外乱あり, Ki=5 のとき)



図9 PID制御の結果 (外乱あり, Ki=5, Kd=1 のとき)

ただし、各所に見られる振動はギアの変化による影響と見られる。外乱の変化が激しいほどギア段の変化も激しくなり、振動を与えることなく安定した一定の速度を保つようにするためには、坂道を走るときエンブレキをかけるというように坂道対象の制御システムを設計する必要があると考える。

7. あとがき

本稿では乗り心地を考慮した車両制御システムを提供するため、フィードバック制御系の P 制御・PI 制御・PID 制御において、各パラメータ K_p , K_i , K_d の数値及び車両速度の上下限の数値の調整方法について研究し、シミュレーションにより検証・評価した。PI 制御と PID 制御については、D 動作を加えることによって少しでも波長の振動を抑えることができるため、どのような制御対象においても PID 制御を使用することが望ましいことを確認した。ただし、PID 制御は PI 制御に比べて位相の遅れが生じることがデメリットである。外乱がある場合は、PID 制御において外乱がない場合よりも偏差が生じやすくなるため I 動作を大きくする必要がある。

今後は、各パラメータ K_p , K_i , K_d の数値の調整において、最適な数値を算出する方法としてジューゲラ・ニコルスの 2 公式と呼ばれる限界感度法と過渡応答法の 2 つの方法についてさらに研究し、本稿のシミュレーション評価との差異について検討する所存である。

参考文献

- [1]佐藤和也,平元和彦,平田研二,“はじめての制御工学”,講談社(2010).
- [2]佐伯正美,“制御工学 –古典制御からロバスト制御へ–”,朝倉書店(2013).
- [3]森泰親,“制御工学”,コロナ社(2001).
- [4]山本重彦,加藤尚武,“PID制御の基礎と応用”,朝倉書店(2005).
- [5]久保孝行,“自動車業界 MBD エンジニアのための Simulink 入門”,TechShare(2012).