

セミナー用テンプレート

動的システム・ロボティクス研究室 B4 田辺 裕翔

2025 年 10 月 21 日

1 はじめに

先月は状態推定についてのまとめを行った。今回は、まとめ直したことによって出てきた問題点の改善を行った。次に今後の方針についての考えを軽く話す。

2 状態推定

前回の状態推定の結果を Fig.2.1~2.5 に示す。紫の実線が UVA/Pdova の応答、青の実線が ssogmm の応答、赤のドットが EKF の推定結果、黄色のドットがオブザーバによる推定結果を表している。拡張カルマンフィルタによる推定結果に対してオブザーバの推定結果は状態 X についてのみ誤推定が発生している。この問題点についてはオブザーバゲインが思っていたよりもバランスが悪くなっていただけなので、そこを修正した。またもう一つ問題点が見つかった。拡張カルマンフィルタのコードにミスがあった。転置しなければならない行列を一つ転置していなかった。コードを修正してシミュレーションをすると、オブザーバと同じように状態 X についてのみ誤推定が出た。そのため拡張カルマンフィルタについても重みの調整を行った。これ以上は細かい調整になりそうだとここで、調整を終えて、シミュレーションをした結果が Fig.2.6~2.10 に示す通りである。青の実線が UVA/Padova の応答、紫の実線が ssogmm の応答、赤のドットがオブザーバの推定結果、黄色のドットが EKF による推定結果を表している。全ての状態がうまく収束していることがわかる。またオブザーバに対して拡張カルマンフィルタの方が収束が早いことがわかる。シミュレーション全体 (4 日間) での RMSE と 4 日目のみの RMSE を Table.2.1 に示す。RMSE は UVA/Padova T1DM シミュレータから得られた血糖値に対する状態推定誤差である。この Table.2.1 から拡張カルマンフィルタによる状態推定結果の方がオブザーバよりも精度が良いことがわかる。今後の mpc の実装を考えても状態の収束は早い方が都合が良いため、状態推定器としては拡張カルマンフィルタの方がいい。

Table2.1: RMSE

オブザーバ	#001	#002	#003	#004	#005	#006	#007	#008	#009	#010
4 日間	0.3890	0.2186	0.5007	0.5859	0.5930	0.6086	0.8343	0.8287	0.6104	0.3509
4 日目のみ	0.2099	0.1938	0.1530	0.1924	0.2173	0.3454	0.2667	0.2152	0.1951	0.1960
EKF	#001	#002	#003	#004	#005	#006	#007	#008	#009	#010
4 日間	0.3437	0.1155	0.4785	0.5575	0.5613	0.4570	0.7989	0.8063	0.5794	0.2899
4 日目のみ	0.1186	0.1147	0.0792	0.1071	0.1192	0.1756	0.1351	0.1266	0.1076	0.1163

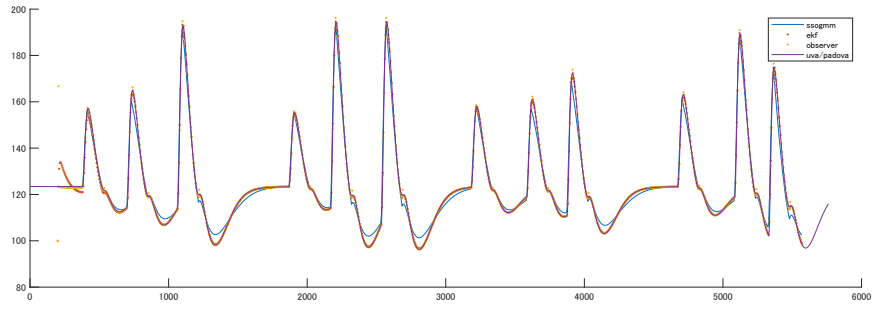


Fig.2.1: G

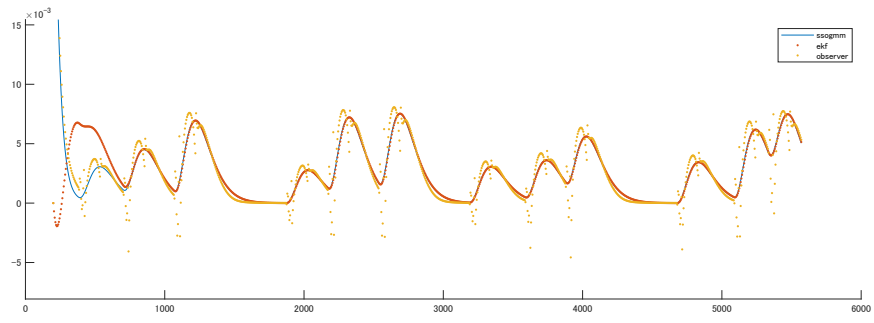


Fig.2.2: X

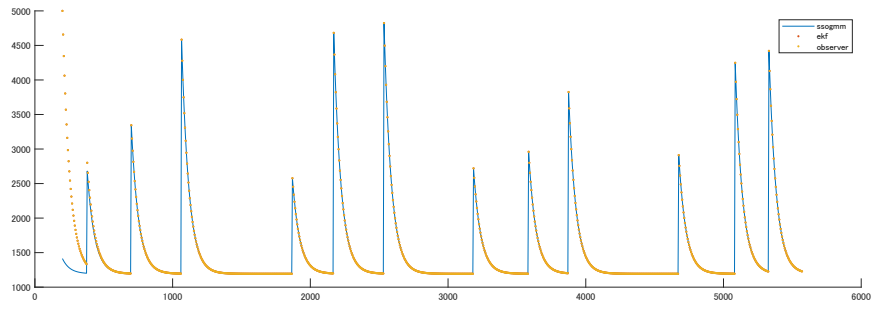


Fig.2.3: I1

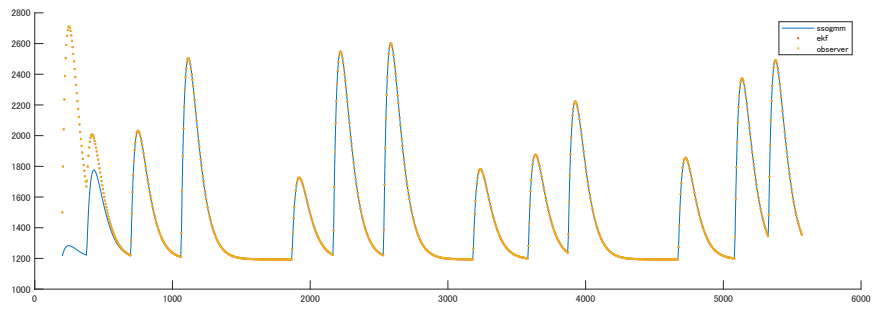


Fig.2.4: I2

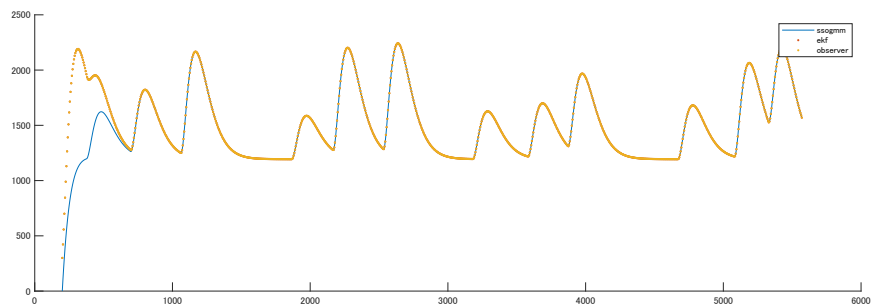


Fig.2.5: Ip

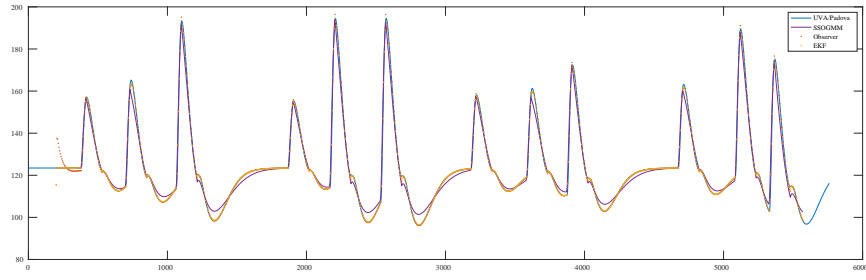


Fig.2.6: G

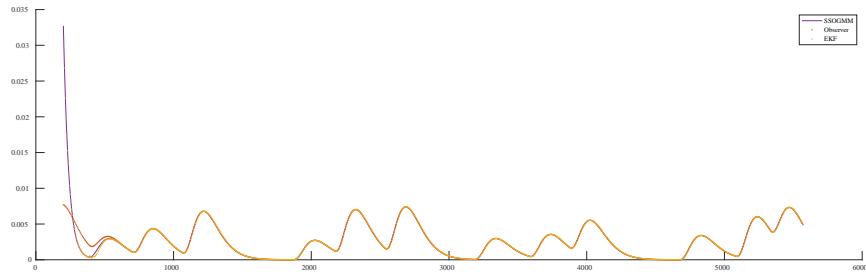


Fig.2.7: X

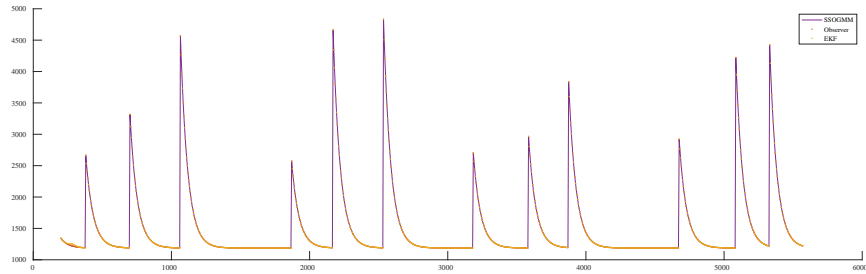


Fig.2.8: I1

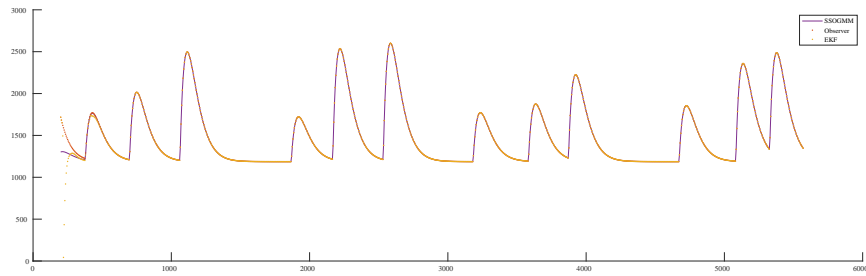


Fig.2.9: I2

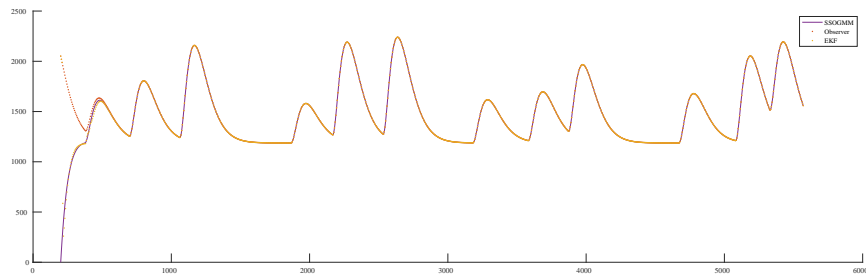


Fig.2.10: Ip

3 今後の方針

三輪さんから過去に作った mpc のコードの場所を教えていただいたので、そちらのコードを参考に現在の環境に合わせて、つまり SSGMM を mpc の予測モデルとして利用できるようにコードを書き換え、とりあえず動作するところまでの準備はできている。最終的には Uva/Padova T1DM シミュレータに mpc を実装し、シミュレーションを行っ

ていきたいが、モードをどう与えたらいいかなど、手間取ることが多そう。なので、最初は matlab 上に実装してある Uva/Padova モデルを使って mcp のシミュレーションを行おうと思う。

また状態推定についても、今までモードについてはすべてわかっているとして考えてきたが、実際にはこのモードについても推定する必要があるためこの方法についても、今後考えていく必要がある。

mpc に関するブロック線図を書く