FAKULTA ELEKTROTECHNICKÁ ZÁPADOČESKÉ UNIVERZITY V PLZNI

Software

Software library of stochastic filtering algorithms

```
function [P]=pmsm_init(variant)
if nargin<1
    variant = 1;
end

% symbolic variables
syms id iq om th ua ub ia ib real
syms Rs Lsd Lsq Fmag J B positive

% rotation to appropriate frame
Rot2dq = [cos(th), sin(th); -sin(th), cos(th)];
Rot2ab = [cos(th), -sin(th); sin(th), cos(th)];

uab= [ua;ub];
udq= Rot2dq*uab;
ud = udq(1);
uq = udq(2);
%% continuous time equations
dx = [1/Lsd*(-Rs*id + Lsq*iq*om + ud);...
    1/Lsq*(-Rs*iq -Lsd*id*om - Fmag*om+uq);...
    1/J*(3/2*4^2*((Lsd-Lsq)*id*iq+Fmag*iq))-B*om;...
    om];

%% discrete
syms dt
x = [id iq om th].';
fx = x+dt*dx; % Euler - can be replaced by expm()
par = [Rs Lsd Lsq Fmag J B dt];

switch variant
    case 1 % classical full dq model
        %% observation
        iab= Rot2ab*[id;iq];

        P.x = x;
        P.u = uab;
        P.par = par;
        P.fx = fx;
        P.hx = iab;
        P.A  = jacobian(P.fx,x);
        P.C  = jacobian(P.hx,x);
        P = export_P(P,'pmsm_dq');
    case  2 % reduced order dq model
        P.x = [om th];
        P.u = [uab;id;iq];
        P.par = par;
        P.fx = fx(3:4);
        P.hx = Rot2ab*fx(1:2);
        P.A  = jacobian(P.fx,P.x);
        P.C  = jacobian(P.hx,P.x);
        P = export_P(P,'pmsm_dqr');
    case 3 % estimator in dq
        %% estimator
        P.x = [Rs Lsd Lsq Fmag J B]';
        P.u = uab;
        P.par = [x;dt];
```

```
void kalman(float om_m, float iqyd){
  //prediction
  if (p11<p11max) {p11=p11+qj;}
  //p12=p12;
  if (p22<p22max) {p22=p22+qT;}


  // C matrix
  c1 = cc1*iqyd-cc2*xT;
  c2 = -cc2*xj;

  // Ry
  r11=(c11*p11+c12*p12)*c11+(c11*p12+c12*p22)*c12+r;
  // inv Ry
  ir11=1/r11;
  //K = P*Ct*iRy;
  k11=ir11*(c11*p11+c12*p12);
  k21=ir11*(c11*p12+c12*p22);
  //P = P- K*C*P
  p11=p11-k11*c11*p11-k11*c12*p12;
  p12=p12-k11*c11*p12-k11*c12*p22;
  p22=p22-k21*c11*p12-k21*c12*p22;

  // correction
  ydif = om_m - (om + xj*c1); // - (om_prediction)
  xj = xj + k11*ydif;
  xT = xT + k21*ydif;
```

- V souladu s platnou metodikou Úřadu vlády ČR je uplatňován software „Software library of stochastic filtering algorithms“.
- Software vznikl v přímé souvislosti s řešením projektu TAČR TE02000103 (CIDAM).
- Knihovna algoritmů pro stochastickou filtraci stavu a parametrů PMSM, včetně generátoru kódu v matlabu. Vyvinutý generátor v jazyce Matlab umožňuje generovat ze zadaného matematického modelu kód pro stochastickou filtraci. Modul využívá symbolický toolbox a generuje kód v jazyce ANSI C. Umožňuje generovat kód pro filtrační algoritmy EKF, UKF a particle filter v podobě stavového filtru nebo sdruženého odhadu stavu a parametrů. Modely generují funkce, které mohou být sdíleny mezi jednotlivými filtry. Zvláštní pozornost je věnována minimální výpočetní náročnosti výsledného kódu. Tento nástroj umožňuje rychle vygenerovat nový stochastický filtr ze zadaných rovnic, přeložit jej pro signálový procesor a otestovat vlastnosti nového filtru v reálném čase. V knihovně jsou nagenerovány filtry pro různé varianty modelů synchronního pohonu s permanentními magnety jak vnitřními tak povrchovými. Jsou připraveny i modely s rozšířením na odhad vířivých proudů a složitějších modelů indukčnost.

EVIDENČNÍ ČÍSLO:

22190-SW005-2018

KONTAKTNÍ OSOBA:

Václav Šmídl

tel.: 37763 4127

vsmidl@rice.zcu.cz

ŘEŠITELSKÉ PRACOVIŠTĚ:

Západočeská univerzita v Plzni

Fakulta elektrotechnická

RICE

Univerzitní 8, 306 14 Plzeň

TA ČR Program Centra kompetence

CIDAM Centrum inteligentních pohonů a pokročilého řízení strojů