|  |  |
| --- | --- |
| *voenmeh* | МИНОБРНАУКИ РОССИИ  федеральное государственное бюджетное образовательное учреждение  высшего профессионального образования  **«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»**  **(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова»)** |
| БГТУ.СМК-Ф-4.2-К5-01 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Факультет |  | А |  | Ракетно-космической техники |
|  |  | шифр |  | наименование |
| Кафедра |  | А5 |  | Динамика и управление полетом летательных аппаратов |
|  |  | шифр |  | наименование |
| Дисциплина |  | Оценка состояния и параметров летательного аппарата | | |

Практическая работа №3

на тему

|  |
| --- |
| Построение непрерывного фильтра Калмана |
| для системы стабилизации угла крена |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы | | | | | А571 |
| Анкудинов А.Н. | | | | | |
| Фамилия И.О. | | | | | |
| **Преподаватель** | | | | | |
|  | |  |  | | |
| Фамилия И.О. Подпись | | | | | |
| Оценка |  | | |  | |
| «\_\_\_\_\_» |  | | | 2021 г. | |

Санкт-Петербург

2021 г.

*Цель работы* – осуществить оценку вектора состояния контура стабилизации угла крена по зашумленным измерениям угла крена с помощью непрерывного фильтра Калмана.

*Сведения из теории*

Динамика системы стабилизации угла крена ЛА определяется следующей системой уравнений:

|  |  |
| --- | --- |
|  | (1) |

где - угол и угловая скорость крена ЛА, - угол отклонения элеронов, – шум системы.

Структурная схема контура стабилизации угла крена представлена на рисунке 1.

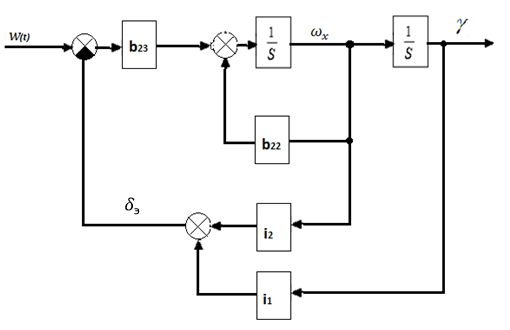


Рисунок 1 – Структурная схема контура стабилизации угла крена

Для реализации алгоритма непрерывной фильтрации Калмана необходимо представить систему (1) в виде:

|  |  |
| --- | --- |
|  | (2) |

где – вектор состояния системы, – вектор возмущения, – вектор управления, – матрица состояния, – матрица возмущения, – матрица управления.

Для системы (1) матрицы , , равны:

|  |  |
| --- | --- |
|  | (3) |

Уравнение измерений необходимо представить в виде

|  |  |
| --- | --- |
|  | (4) |

где – вектор измерения, – вектор ошибки измерения, – матрица измерения.

В данной системе стабилизации угла крена измерению подлежит лишь угол крена, тогда для системы (1) уравнение (4) можно представить в виде:

|  |  |
| --- | --- |
| *.* | (5) |

Матрица измерения для уравнения (5) равна:

|  |  |
| --- | --- |
|  | (6) |

Алгоритм оптимального фильтра Калмана для непрерывного времени:

1. Оптимальная оценка для системы (2), (4) удовлетворяет уравнению

|  |  |
| --- | --- |
|  | (7) |

где – матрица передачи фильтра.

1. Матрица определяется с помощью соотношения

|  |  |
| --- | --- |
|  | (8) |

где – корреляционная матрица ошибки фильтрации

1. Корреляционная матрица , удовлетворяет матричному дифференциальному уравнению

|  |  |
| --- | --- |
|  | (9) |

с начальным условием .

Структурная схема фильтра Калмана представлена на рисунке 2.

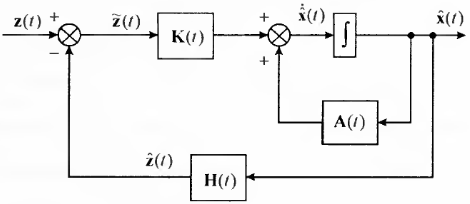


Рисунок 2 – Структурная схема непрерывного фильтра Калмана

Для реализации алгоритма (7) - (9) необходимо знать матрицы интенсивностей входных шумов системы и шумов измерений . Они будут равны дисперсиям ошибок системы и измерений соответственно:

|  |  |
| --- | --- |
|  | (10) |

Для того, чтобы интегрировать систему уравнений (9) необходимо знать начальную матрицу . Как правило, начальная корреляционная матрица задается в виде диагональной матрицы с дисперсиями шума системы на главной диагонали:

*Начальные данные*

Таблица 1 – Исходные данные

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Вариант* |  |  |  |  |
| 1 | -0.55 | 3 | 0.8 | 1 |

*Ход работы*

Для решения поставленной задачи был написан скрипт в среде *Matlab.*

Моделирование производилось с шагом 0.1с в интервале времени от 0 до 20 с.

Начальные данные для моделирования системы представлены в таблице 2.

Таблица 2 – Начальные данные для моделирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ,  град/с | ,  град | ,  град | , град/с | ,  град | , град/с |
| 0.02 | 0.5 | 5 | 0 |  | 0 |

где – измерение, полученное в начальный момент времени.

Результаты моделирования представлены на рисунках 3-4.



Рисунок 3 – Графики измерения, реального значения и оценки угла крена во времени



Рисунок 4 – Графики реального значения и оценки угловой скорости крена во времени

По рисункам 3, 4 видно, что оценки и сходятся к истинным значениям.

*Вывод:* был оценен вектор состояния контура стабилизации угла крена во времени по зашумленным измерениям угла крена с помощью оптимального непрерывного фильтра Калмана. Таким образом, рассчитана оценка всего вектора лишь по одному измерению. Полученные оценки довольно точно приближают реальные значения.

*Листинг скрипта Matlab*

clear; clc; close all;

global i1 i2 sig\_syst sig\_meas A B G H R Q h

%% Параметры системы

b22 = -0.55;

b23 = 3;

i1 = 0.8;

i2 = 1;

%% СКО шумов

sig\_syst = 0.02;

sig\_meas = 0.5;

%% Матрицы

A = [ 0 1;

0 b22 ];

B = [ 0;

b23 ];

G = B;

H = [1 0];

Q = sig\_syst^2;

R = sig\_meas^2;

%% Параметры моделирования и вектор времени t

h = 0.1;

t\_end = 20;

n = t\_end / h + 1;

t = 0 : h : t\_end;

%% Н.У.

% начальное значение корреляционной матрицы ошибок

P = [ sig\_syst^2 0 ;

0 sig\_syst^2 ];

% начальный вектор состояния

z\_real(:, 1) = [ 5; 0 ];

% начальное измерение угла крена

hi(1) = z\_real(1, 1) + normrnd(0, sig\_meas);

% начальная оценка вектора состояния: принимаем для крена - равной нач. измерению, для угл. скорости - нулю

z\_est(:, 1) = [ hi(1); 0 ];

%% Решение

for i = 2 : 1 : n

% модель оценок вектора состояния z

z\_est(:, i) = rungeKuttaStep( @dzEst\_dt, z\_est(:, i-1), P, hi(i-1) );

% модель реального вектора состояния z

z\_real(:, i) = rungeKuttaStep( @dzReal\_dt, z\_real(:, i-1), 0, 0 );

% система ДУ Риккати

P = rungeKuttaStep( @dP\_dt, P, 0, 0 );

% текущее измерение угла крена

hi(i) = z\_real(1, i) + normrnd(0, sig\_meas);

end

%% Графики

figure; % крен

title('\gamma(t)')

hold, grid on

plot( t, hi, 'c', 'DisplayName', 'измерение' );

plot( t, z\_real(1, :), 'm', 'DisplayName', 'реальность' );

plot( t, z\_est(1, :), 'b', 'markersize', 3, 'DisplayName', 'оценка' );

legend show

figure; % угл. скорость крена

title('\omega(t)')

hold, grid on

plot( t, z\_real(2, :), 'm', 'DisplayName', 'реальность' );

plot( t, z\_est(2, :), 'b', 'markersize', 3, 'DisplayName', 'оценка' );

legend show

%% Функции

function RETURN = dzEst\_dt( z\_est, P, hi )

global R A H B i1 i2

K = P \* H' \* R^(-1);

u\_est = -i1\*z\_est(1) - i2\*z\_est(2);

RETURN = A\*z\_est + B\*u\_est + K\*( hi - H\*z\_est );

end

function RETURN = dP\_dt( P, unused1, unused2 )

global R Q A G H

RETURN = A\*P + P\*A' - P \* H' \* R^(-1) \* H \* P + G \* Q \* G';

end

function RETURN = dzReal\_dt( z, unused1, unused2 )

global i1 i2 sig\_syst A B G

u = -i1\*z(1) - i2\*z(2);

RETURN = A\*z + B\*u + G\*normrnd(0, sig\_syst);

end

function RETURN = rungeKuttaStep(f, z\_curr, param1, param2)

% params may be unused or used in some of the functions

global h

k1 = f(z\_curr, param1, param2);

k2 = f(z\_curr + h\*k1/2, param1, param2);

k3 = f(z\_curr + h\*k2/2, param1, param2);

k4 = f(z\_curr + h\*k3, param1, param2);

RETURN = z\_curr + h\*(k1 + 2\*k2 + 2\*k3 + k4) / 6;

end