Переписать М-функц из Matlab на С++

function [S\_out, Error] = ElementPOI\_OF(S\_in, KoeffOF, InTime)

% Функция реализует согласованную фильтрацию

%

% Входные данные:

% S\_in - массив входного сигнала, 1 такт - комплексные числа;

% размерность (х1, 1), где х1 - длина дистанции, квант (команда)

% KoeffOF - массив коэффициентов оптимального фильтра - комплексные

% числа; размерность (х2, 1), где х2 - количество отсчетов ДПФ

% или количество отсчетов оптимального фильтра во временной

% области

% InTime - флаг, поднятое состояние которого указывает, что

% коэффициенты оптимального фильтра, передаваемые функции, заданы

% во временной области. Если коэффициенты оптимального фильтра

% заданы в частотной области, то переменной InTime должно быть

% присвоено значение 0.

%

% Выходные данные:

% S\_out - массив выходного сигнала, 1 такт - комплексные числа;

% размерность (х1, 1)

% Error - переменная, содержащая номер ошибки, если таковая есть -

% целое число

%

Error = 0;

S\_in\_Size = size(S\_in);

KoeffOF\_Size = size(KoeffOF);

S\_out = zeros(S\_in\_Size(1), 1);

% Проверяем, нет ли ошибок во входных данных

if S\_in\_Size(2) ~= 1

Error = 3;

end

if KoeffOF\_Size(2) ~= 1

Error = 4;

end

if Error < 3

if InTime == 0

% Проверяем, является ли количество коэффициенто оптимального фильтра

% степенью двух

Step2 = 0;

while 2^Step2 < KoeffOF\_Size(1)

Step2 = Step2 + 1;

end

if 2^Step2 ~= KoeffOF\_Size(1)

Error = 1;

end

% Проверяем, не больше ли длина дистанции количества коэффициентов

% оптимального фильтра

if S\_in\_Size(1) > KoeffOF\_Size(1)

Error = 2;

end

KolDPF = KoeffOF\_Size(1);

else

% Находим необходимое число отсчетов ДПФ

Step2 = 0;

while 2^Step2 < S\_in\_Size(1)

Step2 = Step2 + 1;

end

KolDPF = 2^Step2;

KoeffOF = fft(KoeffOF, KolDPF);

end

% Реализация оптимального фильтра

S\_in\_DPF = fft(S\_in, KolDPF); % спектр сигнала на входе оптимального фильтра

S\_out\_DPF = zeros(KolDPF, 1); % спектр сигнала на выходе оптимального фильтра

for IndexDPF = 1 : KolDPF

S\_out\_DPF(IndexDPF, 1) = S\_in\_DPF(IndexDPF, 1) \* KoeffOF(IndexDPF, 1);

end

S\_out\_Long = ifft(S\_out\_DPF, KolDPF);

% Выравнивание массивов входных и выходных данных

for NumKvant = 1 : S\_in\_Size(1)

if NumKvant <= KolDPF

S\_out(NumKvant, 1) = S\_out\_Long(NumKvant, 1);

end

end

end

return