```
function [ref_out, accum_out, state_out] = pll(ref_in, state_in);

% [ref_out] = pll(ref_in, state_in);
% Does PLL tracking of the input waveform.  Operates on complete waveform.
% Inputs:
%   ref_in      Input reference
%   state_in    State and parameters
% Outputs:
%   ref_out     Output reference
%   accum_out   Output accumulator

% Get parameters
state = state_in;

f0 = state.f0;
K = state.K;
a = state.a;
b = state.b;

N = length(ref_in);

ref_out = zeros(N, 1);
accum_out = zeros(N, 1);
%% Estimate amplitude of block
amp_est = mean(abs(ref_in))*(pi/2);
%% Get accumulator
accum = state.accum;
%% PLL
for n=1:N,
  % Multiply
  z(n) = state.ref_in_prev * state.ref_out_prev / amp_est;
  % Loop filter
  v(n) = state.a(1)*state.v_prev + state.b(1)*z(n) + state.b(2)*state.z_prev;

  state.z_prev = z(n);
  state.v_prev = v(n);
  % VCO
  state.accum = state.accum + state.f0 - state.K*v(n)/(2*pi);
  state.accum = state.accum - floor(state.accum);

  accum_out(n) = state.accum;
  ref_out(n) = sin(2*pi*state.accum);
  state.ref_out_prev = ref_out(n);
  state.ref_in_prev = ref_in(n);
end
state_out = state;
```