A Apêndice

Programa compilado no ambiente Matlab para realizar o cálculo dos modelos linear e angular bem como o projeto dos ganhos dos controladores linear e angular que serão utilizados na simulação no ambiente Simulink.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% VARIAVEIS ELETRICAS E MECANICAS %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

m = 0.373375;       % MASSA DO ROBO
r_d = 0.015;        % RAIO DA RODA DIREITA
r_e = 0.015;        % RAIO DA RODA ESQUERDA
d = 0.08;           % DIMENSAO ROBO
K_d = 3.3*10^-2;    % CONSTANTE DE TORQUE MOTOR DIREITO
K_e = 3.3*10^-2;    % CONSTANTE DE TORQUE MOTOR ESQUERDO
J_d = 5.1*10^-6;    % MOMENTO DE INERCIA DO ACOPLAMENTO RODA/MOTOR DIREITO
J_e = 5.1*10^-6;    % MOMENTO DE INERCIA DO ACOPLAMENTO RODA/MOTOR ESQUERDO
B_d = 4.78*10^-6;   % COEFICIENTE DE ATRITO DO MOTOR DIREITO
B_e = 4.78*10^-6;   % COEFICIENTE DE ATRITO DO MOTOR ESQUERDO
p_d = 0.006;        % RESISTIVIDADE ELETRICA MOTOR DIREITO
p_e = 0.006;        % RESISTIVIDADE ELETRICA MOTOR ESQUERDO
J = 25.5*10^-6;     % MOMENTO DE INERCIA ROBO
B_t = 0.58*10^-6;   % COEFICIENTE DE ATRITO COM O SOLO EM MOVIMENTO ROTACIONAL
B_l = 0.43;         % COEFICIENTE DE ATRITO COM O SOLO EM MOVIMENTO LINEAR

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%% MATRIZES %%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

J_m = [J_d    0;     % MATRIZ DE INERCIAS DOS ACOMPLAMENTOS RODAS/MOTORES
         0  J_e];

B_m = [B_d    0;     % MATRIZ DE COEFICIENTES DE ATRITO DOS MOTORES
         0  B_e];

J_r = [m   0;        % MATRIZ DE MASSA E INERCIA ROBO
       0   J];

B_r = [B_l    0;     % MATRIZ DE COEFICIENTES DE ATRITO SOLO/MOVIMENTOS
         0  B_t];

K_m = [K_d    0;     % MATRIZ DE CONSTANTES DE TORQUE DOS MOTORES
         0  K_e];

p = [p_d     0;      % MATRIZ DE RESISTIVIDADE ELETRICA DOS MOTORES
       0   p_e];

w_T_u = [1/r_d   d/(2*r_d);   % MATRIZ DE CONVERSAO DE VELOCIDADES DO REFERENCIAL DO ROBO
         1/r_e  -d/(2*r_e)];  % PARA O REFERENCIAL DOS ATUADORES

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%% MATRIZES MODELO %%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% PARTE ELETRICA
K_u = w_T_u' * p * K_m;
% PARTE MECANICA
M_u = J_r + w_T_u' * J_m * w_T_u;
% RESTRICOES ATRITO/ACIONAMENTO
B_u = B_r + w_T_u' * [p * K_m * K_m + B_m] * w_T_u;
% MATRIZ CONVERSAO SINAL DE CONTROLE ROT/LIN <-> TENSAO NOS MOTORES
T = -inv(-inv(M_u)*B_u)*inv(M_u)*K_u;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%% MATRIZES MODELO CINEMATICO %%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% MODELO EM SS  X_p = A*X + B*U
A = [-inv(M_u)*B_u  zeros(2);
        eye(2)      zeros(2)];

B = [inv(M_u)*K_u ; zeros(2)] * inv(T);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%% MODELOS %%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

s = tf('s');

G_s = B(1,1) / (s^2 + B(1,1)*s) % MODELO LINEAR A PARTIR DE B
G_t = B(2,2) / (s^2 + B(2,2)*s) % MODELO ROTACIONAL A PARTIR DE B


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%% CONTROLADOR LINEAR %%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Ta = 3.6;               % TEMPO DE ACOMODACAO DESEJADO
zeta = 1.1;             % ZETA PRA SISTEMA SUPERARMOTECIDO
Wn = 4 / (zeta*Ta);     % FREQ. NAT. DE OSC. DESEJADO

Kd_s = (2*zeta*Wn - B(1,1)) / B(1,1); % GANHO DERIVATIVO
Kp_s = (Wn*Wn)/B(1,1);                % GANHO PROPORCIONAL

C_s = Kp_s + Kd_s*s;                  % CONTROLADOR PD LINEAR

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%% CONTROLADOR ANGULAR %%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Ta = 1.5;                       % TEMPO DE ACOMODACAO DESEJADO
OV = 0.0001;                    % MAXIMO PICO DESEJADO
x = log(OV)^2;
zeta = sqrt ( x / (x + pi^2));  % ZETA PARA Ta e OV DESEJADOS
Wn = 4 / (zeta*Ta);             % FREQ. NAT. DE OSC. DESEJADO

Kd_t = (2*zeta*Wn - B(2,2)) / B(2,2); % GANHO DERIVATIVO
Kp_t = (Wn*Wn)/B(2,2);                % GANHO PROPORCIONAL

C_t = Kp_t + Kd_t*s;                  % CONTROLADOR PD ANGULAR
```

Programa 1: Código utilizado para realizar o cálculo dos ganhos dos controladores PD.

Programa utilizado junto a estrutura descrita na Figura 14 para realizar a seleção de referência que o robô seguirá.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%% SELECAO DE TRAJETORIA %%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Ref = fcn(sel,cir,lis,mhs)

if(sel == 1)        % SELECIONA CIRCULAR
    Ref = cir;
elseif(sel == 2)    % SELECIONA CURVA DE BERNOULLI
    Ref = lis;
else                % ESTRATEGIA DE DEFESA MHS
    Ref = mhs;
end
```

Programa 2: Código utilizado para selecionar a referência.

Programa utilizado junto a estrutura descrita na Figura 15 para realizar o cálculo do ângulo de referência (ϕ) e do deslocamento (Δl) definidos respectivamente pelas equações (14) e (16).

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%% FUNCAO CALCULO DE phi E dl %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [phi,dl] = fcn(Xd,Yd,X,Y)

phi = atan2((Yd-Y),(Xd-X)); % CALCULO DE PHI
dl = sqrt((Yd-Y)*(Yd-Y)+(Xd-X)*(Xd-X)); % CALCULO DE DL
end
```

Programa 3: Código utilizado para cálculo dos valores de ϕ e Δl.

Programa utilizado junto a estrutura descrita na Figura 15 para realizar o cálculo dos erros linear (e_s) e angular (e_θ) definidos respectivamente pelas equações (15) e (17).

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%% CALCULO DOS ERROS LINEAR E ANGULAR %%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [es,etheta] = fcn(phi,theta,dl)

dphi = phi-theta;
es = dl*cos(phi-theta); % ERRO LINEAR

    if(abs(dphi) < pi/2) % ERRO ANGULAR
        etheta = dphi;
    else
        etheta = dphi - pi*sign(dphi);
    end
end
```

Programa 4: Código utilizado para cálculo dos valores de e_s e e_θ.

Programa utilizado junto a estrutura descrita na Figura 16 para realizar a saturação dos sinais de tensão e_d e e_e sobre os motores que acionam as rodas do robô.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%% CONVERSAO LIN/ROT -> TENSAO MOTORES %%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [ed,ee] = fcn(ev,ew)

    ed = 20.0973*ev + 0.1529*ew; % TENSAO MOTOR DIREITO
    ee = 20.0973*ev - 0.1529*ew; % TENSAO MOTOR ESQUERDO

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%% CONVERSAO TENSAO MOTORES -> LIN/ROT %%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [ev,ew] = fcn(ed,ee)

    ev = 0.0249*ed + 0.0249*ee; % SINAL CONTROLE LINEAR
    ew = 3.2695*ed - 3.2695*ee; % SINAL CONTROLE ROTACIONAL
end
```

Programa 5: Código utilizado para realizar a saturação dos sinais de controle nos motores.

Programa utilizado junto a estrutura descrita na Figura 17 para realizar a converção do ângulo de orientação do robô (θ) para o intervalo em que o ângulo de referência (ϕ) é gerado.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%% CONVERSAO DE THETA %%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function thetao = fcn(thetai)

if(thetai >= 0) % VERIFICA SE O THETA DO ROBO E MAIOR QUE ZERO
    if((2*pi - thetai) < 0) % VERIFICA SE E MAIOR QUE 2PI
        thetai = 2*pi - thetai; % CONVERTE PARA O INTERVALO
        thetai = -thetai;       % DE 0 A 2PI
    end

    if((thetai <= pi) && (thetai >= 0)) % VERIFICA SE THETA ESTA EM PI >= THETA >= 0
        thetao = thetai; % MANTEM THETA NO INTERVALO
    else % THETA ESTA NO INTERVALO PI <= THETA <= 2PI
        thetao = thetai - 2*pi; % CONVERTE THETA PARA O INTERVALO -PI <= THETA <= 0
    end
else % THETA DO ROBO E MENOR QUE ZERO
   if((thetai >= -pi) && (thetai <= 0)) % VERIFICA SE THETA ESTA EM -PI <= THETA <= 0
        thetao = thetai; % MANTEM THETA NO INTERVALO
   else % THETA ESTA NO INTERVALO -PI >= THETA >= -2PI
        thetao = thetai + 2*pi; % CONVERTE THETA PARA O INTERVALO PI >= THETA >= 0
    end
end
```

Programa 6: Código utilizado para realizar a converção do ângulo θ.

Diagrama de blocos utilizado para realizar a simulação no ambiente Simulink, as imagens posteriores demonstram a estrutura de cada bloco.

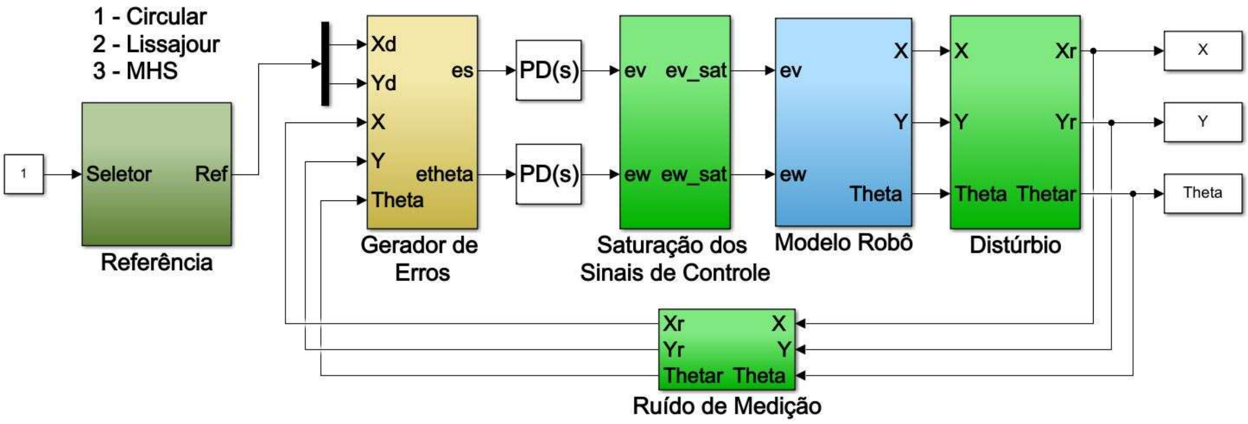

Figura 13: Estrutura utilizada no softaware Simulink para realizar a simulação do robô.

Estrutura responsável por selecionar a referência a ser rastreada pelo robô.

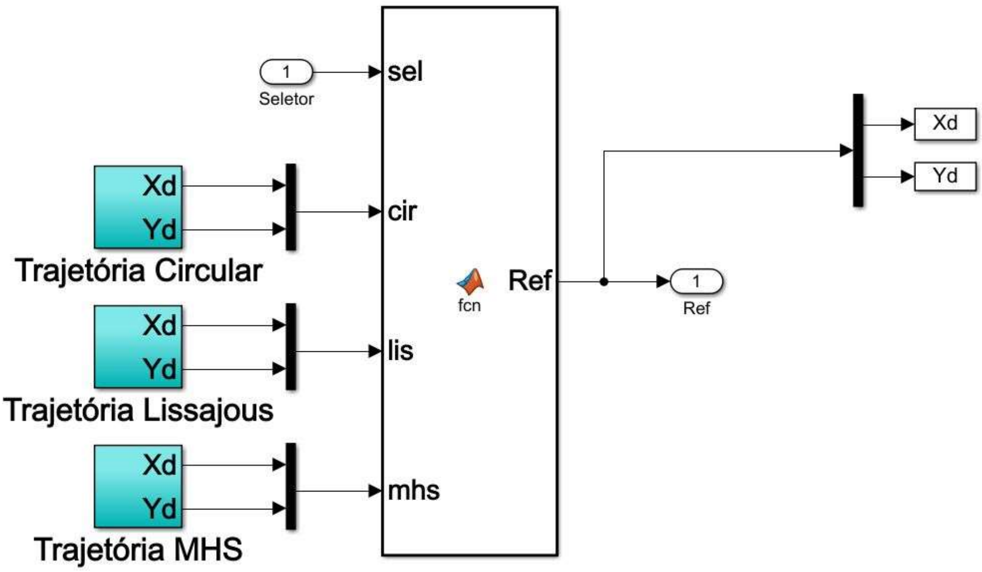

Figura 14: Estrutura utilizada no bloco responsável por selecionar a referência.

Estrutura do bloco utilizado para realizar o cálculo do ângulo de referência (ϕ) e do deslocamento (Δl) definidos respectivamente pelas equações (14) e (16) e também o cálculo dos erros linear (e_s) e angular (e_θ) definidos respectivamente pelas equações (15) e (17).

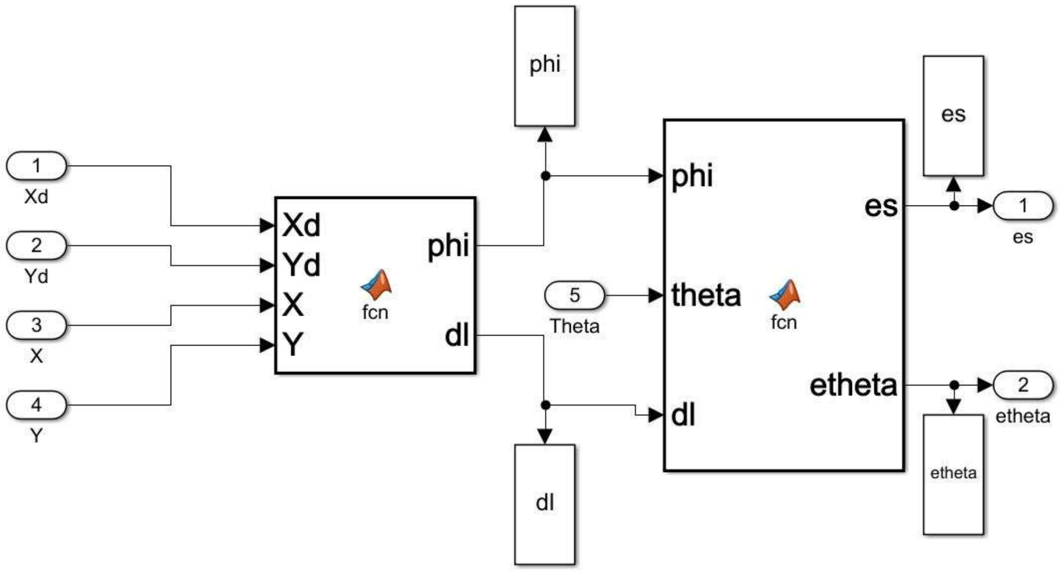

Figura 15: Estrutura utilizada no bloco responsável por calcular os erros linear e angular.

Estrutura do bloco utilizado para realizar a converção dos sinais de controle do formato linear/angular para o formato de tensão nos motores e_d e e_e e saturar estes sinais no intervalo de -6V a 6V, estes sinais devem ser convertidos novamente para o formato linear/angular para serem aplicado aos modelos do sistema.

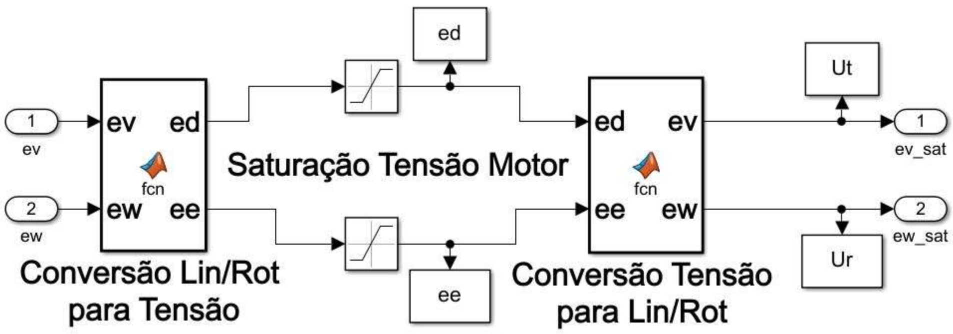

Figura 16: Estrutura utilizada no bloco responsável por saturar os sinais de controle nos motores do robô.

Estrutura do bloco que possui os modelos linear e angular do robô, este bloco recebe os sinais de controle linear e angular e fornece em sua saída os valores de posição (x,y) e orientação (θ) do robô.

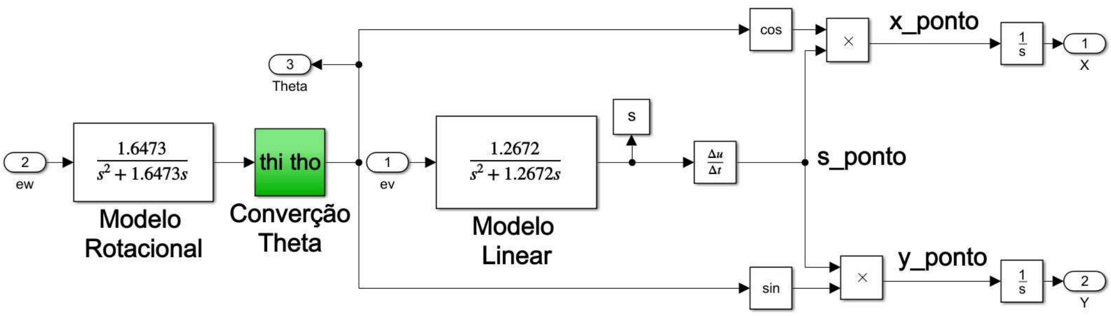

Figura 17: Estrutura utilizada no bloco que simula os modelos linear e angular do robô.

Estutura do bloco utilizado para aplicar um distúrbio de duração de um segundo na saída do sistema depois que o mesmo já se estabilizou.

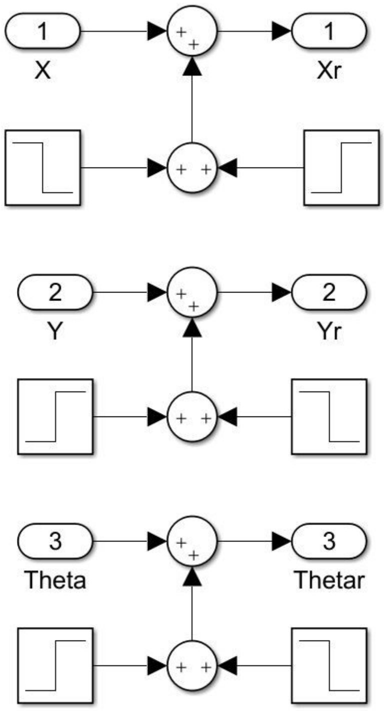

Figura 18: Estrutura utilizada no bloco que insere um distúrbio nas saídas do robô.

Estrutura utilizada para aplicar ruídos de medição na realimentação do sistema, visto que em um sistema real o único modo de visualiazar as saídas do sistema e por meio da medição e dessa forma está sujeita ao efeito de ruído.

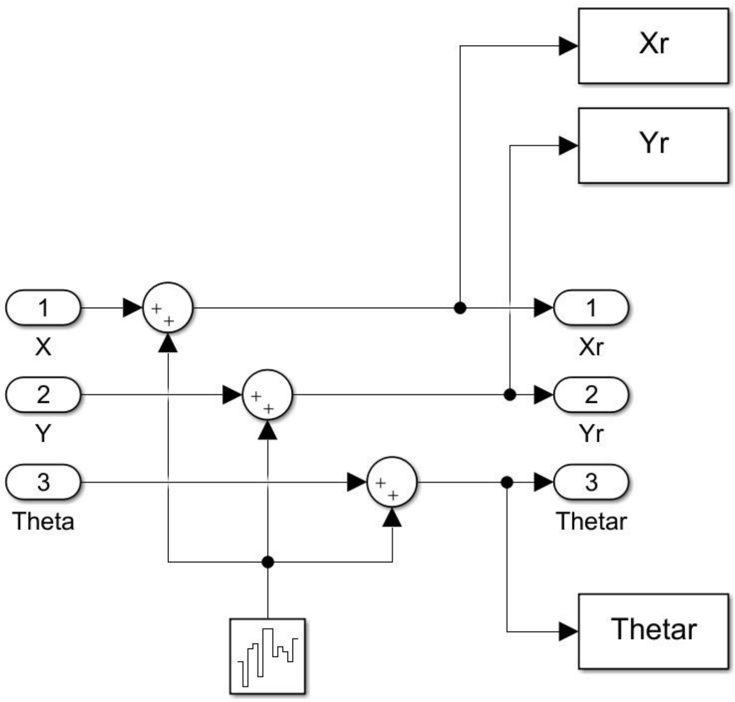

Figura 19: Estrutura utilizada no bloco que insere um ruído de medição na realimentação de posição e orientação do robô.