ANEXO A: Especificaciones del reactor de transesterificación.

En la sección 2.4.2, se detallan las ecuaciones que describen la dinámica del reactor. Para poder simular las ecuaciones, se usaron los datos descritos en este anexo. Las características físicas del reactor son similares a las del reactor usado en [85] y se describen en la Tabla A.1.

Tabla A.1: Características del reactor.

Parámetros	Símbolo	Valor	Unidad
Volumen del reactor	V	27.08	m3
Área total de transferencia de calor	A_H	50.48	m2

Las concentraciones de ingreso del aceite vegetal usado, los coeficientes de las constantes químicas (como la capacidad calorífica, calor de la reacción, constantes cinéticas de la reacción, energía de activación) han sido tomadas de [17]. En las siguientes tablas se muestran los datos usados para la simulación del sistema.

Tabla A.2: Factores pre exponenciales [m^3/ (mol · s)].

k_{o1}	k_{o2}	k_{o3}	k_{o4}	k_{o5}	k_{o6}
5.3992e-04	0.0011	0.0043	0.0037	3.1738e-05	3.9658e-05

Tabla A.3: Energías de activación [J/mol]

E_{a1}	E_{a2}	E_{a3}	E_{a4}	E_{a5}	E_{a6}
5.3992e-04	0.0011	0.0043	0.0037	3.1738e-05	3.9658e-05

Tabla A.4: Capacidades caloríficas [J/ (mol · s)].

C_{pTG}	C_{pDG}	C_{pMG}	C_{pA}	C_{pG}	C_{pE}
86.6864	1791.5150	1298.3897	804.9159	231.0638	609.2845

Tabla A.5: Calor de la reacción [J/ mol].

ΔH_1	ΔH_2	ΔH_3
14869.3	35800.0	-56730.7

Tabla A.6: Masas molares [Kg/ mol].

M_{TG}	M_{DG}	M_{MG}	M_E
873.058e-3	612.737e-3	352.415e-3	292.363e-3

Los valores de concentración de la mezcla de aceite vegetal usado con metanol fueron tomados de [15], y se muestran en la Tabla A.7.

Tabla A.7: Concentraciones de la mezcla de ingreso [mol/m^3].

C_{TGo}	C_{DGo}	C_{MGo}	C_{Ao}	C_{Go}	C_{Eo}
678.57	70.7	45	6.3e+03	0	0

ANEXO B: Archivos de Matlab.

dmc_ucnstr_param.m

```
function [T,H,M,sd,n] = dmc_ucnstr_param(su,sd,p,c,landa)
n = numel(su);
n1 = numel(sd);
if  n1 < n
    sd = [sd;sd(end)*ones(n-n1,1)];
elseif n1 > n
    sd = sd(1:n);
end

G = toeplitz(su(1:p),fliplr(su((p-c+1):p)));
G = tril(G,0);
M_diag = ones(n-1,1);
M1 = diag(M_diag,1);
M = M1;
M(n,n) = 1;
if p > n
    T = [M; zeros(p-n,n)];
    T(n+1:end,n) = 1;
elseif p == n
    T = M;
else
    T = M(1:p,1:n);
end
H = (G'*G+landa*eye(c))\G';
```

dmc_uncstr/DMC

```
function [uk,v_fk_m1,dk] = dmc_ucnstr(uk_1,v_fk,yk,dk,ysp,T,H,M,sd,su,...
                                      p,dk_1)

fk = v_fk(1);
sd_p = sd(1:p);

%% Cálculos on-line (Ikonen, 2013, p. 20)
bk = yk-fk;
d_dk = dk - dk_1;

v_ypk_m1 = T*v_fk+sd_p*d_dk+bk;
v_ek_m1 = ysp-v_ypk_m1;

d_uk = H(1,:)*v_ek_m1;

uk = uk_1+d_uk;

v_fk_m1 = M*v_fk+su*d_uk+sd*d_dk;
```

dmc_cnstr_param.m

```
function [T,H,G,M,A,sd,n] = dmc_cnstr_param(su,sd,pl,c,landa)
n = numel(su);
n1 = numel(sd);
if  n1 < n
    sd = [sd;sd(end)*ones(n-n1,1)];
```

```
elseif n1 > n
    sd = sd(1:n);
end

G = toeplitz(su(1:pl),fliplr(su((pl-c+1):pl)));
G = tril(G,0);
M_diag = ones(n-1,1);
M1 = diag(M_diag,1);
M = M1;
M(n,n) = 1;
if pl > n
    T = [M; zeros(pl-n,n)];
    T(n+1:end,n) = 1;
elseif pl == n
    T = M;
else
    T = M(1:pl,1:n);
end
IL = tril(ones(c),0);
A1 = [eye(c);-eye(c)];
A2 = [IL;-IL];
A3 = [G;-G];
A = [A1;A2;A3];
H = G'*G+landa*eye(c);
```

dmc_cnstr/QDMC

```
function [uk,v_fk_m1,dk] = dmc_cnstr(uk_1,v_fk,yk,dk,ysp,du_max,du_min,...
                                     u_max,u_min,y_max,y_min,T,H,G,M,...
                                     A,sd,su,p,v1_c,v1_p,dk_1)
coder.extrinsic('optimoptions');
coder.extrinsic('quadprog');

vd_uk = v1_c;
fk = v_fk(1);
sd_p = sd(1:p);

%% Cálculos on-line (Ikonen, 2013, p. 20)
bk = yk-fk;
d_dk = dk - dk_1;

v_ypk_m1 = T*v_fk+sd_p*d_dk+bk;
v_ek_m1 = ysp-v_ypk_m1;

%% Modificado de DMC sin restricciones
v_c = -G'*v_ek_m1;

b1 = [du_max*v1_c;-du_min*v1_c];
b2 = [u_max*v1_c;-u_min*v1_c]-[v1_c;-v1_c]*uk_1;
b31 = y_max*v1_p-v_ypk_m1;
b32 = -y_min*v1_p+v_ypk_m1;
b3 = [b31;b32];
v_b = [b1;b2;b3];

options = optimoptions('quadprog','Algorithm','interior-point-convex',...
                       'Display','off');

[vd_uk,fval] = quadprog(H,v_c,A,v_b,[],[],[],[],[],options);

d_uk = vd_uk(1);

uk = uk_1+d_uk;
v_fk_m1 = M*v_fk+su*d_uk+sd*d_dk;
```

dmc_cnstr_soft_param.m

```
function [T,H,G,M,A,sd,n] = dmc_cnstr_soft_param(su,p,c,sd,landa,prho)
n = numel(su);
n1 = numel(sd);
if  n1 < n
    sd = [sd;sd(end)*ones(n-n1,1)];
elseif n1 > n
    sd = sd(1:n);
end

G = toeplitz(su(1:p),fliplr(su((p-c+1):p)));
G = tril(G,0);
M_diag = ones(n-1,1);
M1 = diag(M_diag,1);
M = M1;
M(n,n) = 1;
if p > n
    T = [M; zeros(p-n,n)];
    T(n+1:end,n) = 1;
elseif p == n
    T = M;
else
    T = M(1:p,1:n);
end
IL = tril(ones(c),0);
A1 = [eye(c),zeros(c,1);-eye(c),zeros(c,1)];
A2 = [IL,zeros(c,1);-IL,zeros(c,1)];
A3 = [G,-ones(p,1);-G,-ones(p,1)];
A4 = [zeros(1,c),-1];
A = [A1;A2;A3;A4];
H = [G'*G+landa*eye(c),zeros(c,1);zeros(1,c),prho];
```

dmc_cnstr_soft/QDMC_soft

```
function [uk,v_fk_m1,dk] =
dmc_cnstr_soft(uk_1,v_fk,yk,dk,ysp,du_max,du_min,...
                                u_max,u_min,y_max,y_min,T,H,G,M,...
                                A,sd,su,p,v1_c,v1_p,dk_1)

coder.extrinsic('optimoptions');
coder.extrinsic('quadprog');

vd_uk = [v1_c;1];
fk = v_fk(1);
sd_p = sd(1:p);

%% Cálculos on-line (Ikonen, 2013, p. 20)
bk = yk-fk;
d_dk = dk - dk_1;

v_ypk_m1 = T*v_fk+sd_p*d_dk+bk;
v_ek_m1 = ysp-v_ypk_m1;

%% Modificado de DMC sin restricciones
v_c = [-G'*v_ek_m1;0];

b1 = [du_max*v1_c;-du_min*v1_c];
b2 = [u_max*v1_c;-u_min*v1_c]-[v1_c;-v1_c]*uk_1;
b31 = y_max*v1_p-v_ypk_m1;
b32 = -y_min*v1_p+v_ypk_m1;
```

```
b3 = [b31;b32];
v_b = [b1;b2;b3;0];

options = optimoptions('quadprog','Algorithm','interior-point-convex',...
                       'Display','off');

[vd_uk,fval] = quadprog(H,v_c,A,v_b,[],[],[],[],[],options);

d_uk = vd_uk(1);

uk = uk_1+d_uk;
v_fk_m1 = M*v_fk+su*d_uk+sd*d_dk;
```