

Derivation of Stirred Tank Reactor Optimal Control

Lorenz T. Biegler
 Department of Chemical Engineering
 Carnegie Mellon University
 Pittsburgh, PA, 15213, USA

Consider the liquid-phase stirred tank reactor shown in Figure 1. We assume

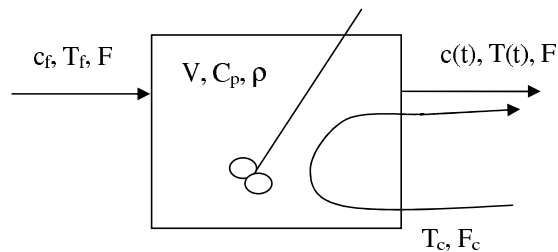


Figure 1: Sketch of Hicks Reactor

constant liquid volume (V), flow (F), density (ρ), heat capacity (C_p), coolant temperature (T_c), coolant flow (F_c), overall heat transfer coefficient ($U(F_c)$), cross sectional area (A_c) and heat of reaction (ΔH), as well as a first order Arrhenius rate law. Following the notation in the figure, the mass and energy balance for the reactor are given by:

$$V \frac{dc}{dt} = F(c_f - c(t)) - V k_{10} \exp(-E/RT) c(t), \quad c(0) = c_{init} \quad (1)$$

$$\begin{aligned} \rho C_p V \frac{dT}{dt} = & F \rho C_p (T_f - T(t)) + \Delta H V k_{10} \exp(-E/RT) c(t) \\ & + U(F_c) A_c (T_c - T), \quad T(0) = T_{init} \end{aligned} \quad (2)$$

We define aggregated parameters $\theta = V/F$, $\alpha u = U A_c / (\rho C_p V)$, $y_f = \rho C_p T_f / \Delta H$, $n = E \rho C_p / (R \Delta H)$, $y_c = \rho C_p T_c / \Delta H$, and redefine the states: $c \leftarrow c/c_f$, $T \leftarrow$

$\rho C_p T / \Delta H$ in order to obtain:

$$\frac{dc}{dt} = (1 - c(t)) / \theta - k_{10} \exp(-n/T) c(t), c(0) = c_{init} \quad (3)$$

$$\frac{dT}{dt} = (y_f - T(t)) / \theta - k_{10} \exp(-n/T) c(t) + \alpha u (y_c - T), T(0) = T_{init} \quad (4)$$

The parameters are given values in the GAMS file below and the optimal control problem is given by:

$$\text{Min} \quad \int_0^{t_f} \alpha_1 (\bar{c} - c(t))^2 + \alpha_2 (\bar{T} - T(t))^2 + \alpha_3 (\bar{u} - u(t))^2 dt \quad (5)$$

$$\text{s.t.} \quad (3) - (4) \quad (6)$$

where \bar{c} , \bar{T} , \bar{u} are the desired values of the states and input and α_j are the desired weights.

\$Title Dynamic Optimization of a Non-Isothermal CSTR

\$OFFUPPER

\$OFFSYMXREF OFFSYMLIST

* option sysout = off;

* OPTION SOLPRINT = OFF;

* This file implements the dynamic optimization of non-isothermal
 * CSTR as proposed by Hicks and Ray and later modified to feature
 * multiple steady-state behaviour. Full discretization of the model
 * and input, output variables renders the problem as a NLP.

*

* More detail on the model can be found in

*

*Flores Tlacuahuac, A., S. Terrazas Moreno, and L. T. Biegler,

* 'On Global Optimization of Highly Nonlinear Dynamic Systems,'

* Industrial and Engineering Chemistry Research, 47, 8, pp 2643 - 2655

*(2008)

Sets i number of finite elements /1*100/
 j number of internal collocation points /1*3/;

Alias (j,k);

```
Scalar  cinit  initial concentration      /0.1367/
        tinit  initial temperature       /0.7293/
        uinit  initial heat transfer rate /390/
        cdes   initial concentration     /0.0944/
        tdes   final temperature         /0.7766/
        udes   final heat transfer rate  /340/
        alpha  dimensionless parameter   /1.95e-04/
        alpha1 dimensionless parameter   /1e+06/
        alpha2 dimensionless parameter   /2e+03/
        alpha3 dimensionless parameter   /1e-03/
        k10    rate constant              /300/
        n      /5/
        cf     /7.6/
        tf     /300/
        tc     /290/
        theta  /20/
        yf     /0.3947/
        yc     /0.3816/
time     /10/
point    /0/
nfe      /100/
ncp      /3/
slopec
slopet
slopeu,
ii,
jj,
point;
```

Table a(j,j) First order derivatives collocation matrix

	1	2	3
1	0.19681547722366	0.39442431473909	0.37640306270047
2	-0.06553542585020	0.29207341166523	0.51248582618842
3	0.02377097434822	-0.04154875212600	0.11111111111111;

Parameter

```
cguess(i,j)
tguess(i,j)
```

```

        ttguess(i,j)
        uguess(i,j)
        h(i)      ;
*
* Initial guess of the decision variables
*
point = 0;
slopec = (cdes-cinit)/(nfe*ncp);
slopet = (tdes-tinit)/(nfe*ncp);
slopeu = (udes-uint)/(nfe*ncp);
for (ii = 1 to nfe,
    for (jj = 1 to ncp,
        point = point+1;
        cguess(i,j) = slopec*point+cinit ;
        tguess(i,j) = slopet*point+tinit ;
        ttguess(i,j) = time*point ;
        uguess(i,j) = slopeu*point+uint ;
    );
);
h(i) = 1/nfe ;

```

```

Variables c(i,j)  concentration
          t(i,j)  temperature
          tt(i,j) time
          u(i,j)  heat transfer rate
          cdot(i,j)
          tdot(i,j)
          c0(i)
          t0(i)
          tt0(i)
          phi      objective function ;

```

```

Equations fobj      criterion definition
          IT, IC, ITT
          FECOLc(i,j)
          FECOLt(i,j)
          FECOLtt(i,j)
          CONc(i)
          CONt(i)

```

```

        CONtt(i)
        ODEc(i,j)
        ODEt(i,j);

fobj..
phi =e= sum((i,j), h(i)*a(j,'3')*(alpha1*(sqr(cdes-c(i,j)))+alpha2*
        sqr(tdes-t(i,j))+alpha3*sqr(udes-u(i,j))));

FECOLc(i,j)$ (ord(i) le nfe).. c(i,j)=e=c0(i)
+time*h(i)*sum(k,a(k,j)*cdot(i,k)) ;

FECOLt(i,j)$ (ord(i) le nfe).. t(i,j) =e= t0(i)
+time*h(i)*sum(k,a(k,j)*tdot(i,k)) ;

FECOLtt(i,j)$ (ord(i) le nfe).. tt(i,j) =e= tt0(i)
+time*h(i)*sum(k,a(k,j)) ;

CONc(i)$ (ord(i) gt 1 and ord(i) le nfe)..
c0(i) =e= c0(i-1) + time*h(i-1)*sum(j, cdot(i-1,j)*a(j,'3'));

CONt(i)$ (ord(i) gt 1 and ord(i) le nfe)..
t0(i) =e= t0(i-1) + time*h(i-1)*sum(j, tdot(i-1,j)*a(j,'3'));

CONtt(i)$ (ord(i) gt 1 and ord(i) le nfe)..
tt0(i) =e= tt0(i-1) + time*h(i-1)*sum(j, a(j,'3'));

ODEc(i,j)$ (ord(i) le nfe).. cdot(i,j) =e= (1-c(i,j))/theta
-k10*exp(-n/t(i,j))*c(i,j) ;

ODEt(i,j)$ (ord(i) le nfe).. tdot(i,j) =e= (yf-t(i,j))/theta
+k10*exp(-n/t(i,j))*c(i,j)-alpha*u(i,j)*(t(i,j)-yc) ;
IC.. c0('1') =e= cinit;
IT.. t0('1') =e= tinit;
ITT.. tt0('1') =e= 0;

*TTT.. time =e= transtime ;

Model hicks /all/;

c.lo(i,j) = 0; c.up(i,j) = 1;

```

```

t.lo(i,j) = 0.1; t.up(i,j) = 1;
u.lo(i,j) = 0; u.up(i,j) = 500;
c0.lo(i) = 0; c0.up(i) = cinit ;
t0.lo(i) = 0.1; t0.up(i)= 1;
c.l(i,j) = cguess(i,j);
t.l(i,j) = tguess(i,j);
tt.l(i,j) = ttguess(i,j);
u.l(i,j) = uguess(i,j);
c0.l(i) = cinit ;
t0.l(i) = tinit ;
cdot.l(i,j) = 1 ;
tdot.l(i,j) = 1 ;

option nlp = coinipopt;
* option nlp = baron;
* option nlp = conopt;
* option nlp = minos;
*hicks.optfile = 1;

Solve hicks minimizing phi using nlp;

display tt0.l, c0.l, t0.l;
* option nlp = conopt;
* Solve hicks minimizing phi using nlp;

```