LUND UNIVERSITY

# Concentration Dynamics of a Time Variable Tank System

## A SIMNON Simulation

Holmberg, Ulf; Wallenborg, Anders

1984

*Document Version:*
Publisher's PDF, also known as Version of record

Link to publication

*Citation for published version (APA):*
Holmberg, U., & Wallenborg, A. (1984). *Concentration Dynamics of a Time Variable Tank System: A SIMNON Simulation*. (Technical Reports TFRT-7280). Department of Automatic Control, Lund Institute of Technology (LTH).

*Total number of authors:*
2

**General rights**
Unless other specific re-use rights are stated the following general rights apply:
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.
• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: https://creativecommons.org/licenses/

**Take down policy**
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

CODEN:LUTFD2/(TFRT-7280)/1-30/(1984)

# CONCENTRATION DYNAMICS OF A TIME VARIABLE TANK SYSTEM. A SIMNON SIMULATION.

ULF HOLMBERG
ANDERS WALLENBORG

DEPARTMENT OF AUTOMATIC CONTROL
LUND INSTITUTE OF TECHNOLOGY
DECEMBER 1984

This report describes a simulation of the concentration dynamics of a time variable tank system. The system considered consists of two cascaded well-mixed tanks, followed by a long tube. The flow through the tanks is a function of time, and hence the resulting dynamic system is time variable.

A mathematical model of the tank system is formulated, and the simulation language SIMNON is used to simulate this model and its properties. SIMNON is also used to study concepts like the adjoint differential equations of the system, and to calculate matrix quantities of particular interest in the analysis of the system, e.g. the reachability and the controllability Gramians.

Special attention is paid to the problems encountered when you try to simulate a variable time delay process with SIMNON.

LUND INSTITUTE OF TECHNOLOGY
DEPARTMENT OF AUTOMATIC CONTROL
Box 118
S 221 00 Lund Sweden

Document name
Report

Date of issue
December 1984

Document number
CODEN:LUTFD2/(TFRT-7280)/1-30/(1984)

Author(s)

Ulf Holmberg

Anders Wallenborg

Supervisor

Sponsoring organization

Title and subtitle

Concentration Dynamics of a Time Variable Tank System.
A SIMNON simulation

Abstract

Key words
Simulation, concentration dynamics, SIMNON, variable time delay.

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title

ISBN

Language
English

Number of pages
30

Recipient's notes

Security classification

DOKUMENTDATABLAD RT 3/81

Distribution: The report may be ordered from the Department of Automatic Control or borrowed through the University Library 2, Box 1010, S-221 03 Lund, Sweden, Telex: 33248 lubbis lund.

# CONCENTRATION DYNAMICS OF A TIME VARIABLE TANK SYSTEM

A SIMNON simulation

Ulf Holmberg
Anders Wallenborg

Department of Automatic Control
Lund Institute of Technology
December, 1984

CONTENTS:

1. Introduction

2. Process description

3. Process model

   3.1 State-space description of the system
   3.2 Input-output description of the system
   3.3 The variable time delay of the tube

4. Simulation of variable time delays

   4.1 Solving implicit algebraic equations in SIMNON
   4.2 A solution method using DAS
   4.3 The standard system DELAY
   4.3 A store-and-search method using DELAY and DAS

5. Simulation of concentration dynamics

   5.1 General
   5.2 Tank system step response
   5.3 Adjoint differential equations
   5.4 Reachability and controllability Gramians
   5.5 Controlling the system from an initial state to zero

6. References

Appendix A - SIMNON source code

Appendix B - Simulation results

# 1. INTRODUCTION

This simulation study is intended to illustrate some of the central results in the first nineteen chapters of Brockett: "Finite Dimensional Linear Systems" [1] (Figures within square brackets indicate references. See chapter 6 for details). The system considered consists of two cascaded well-mixed tanks, followed by a long tube symbolizing a pure time delay. The flow through the tanks is a function of time, and hence the resulting dynamic system is time variable. The input to the system is the concentration of a fictive substance in the flow into the first tank, and we study the dynamics of the concentrations in the two tanks and the concentration after the delay tube.

A mathematical model of the tank system is formulated, and the simulation language SIMNON [2], [3] is used to simulate this model and its properties. SIMNON is also used to study concepts like the adjoint differential equations of the system, and to calculate matrix quantities of particular interest in the analysis of the system, e.g. the reachability and the controllability Gramians.

Special attention is paid to the problems encountered when you try to simulate a variable time delay process with SIMNON.

## 2. PROCESS DESCRIPTION

The process consists of two well-mixed tanks and a long tube. The tanks as well as the tube are filled with a liquid, e.g. water, and they are arranged so that the output of the first tank is connected to the input of the second, and the output of the second tank is connected to the input of the long tube. Since both of the tanks are filled, we have no accumulation of liquid anywhere and consequently the same flow runs through both the tanks and the tube. This flow, however, may be a function of time.

In this simulation study we have assumed the flow , q(t), to have a sinusoidal time variation:

$$q(t) = q_0 + q_1 \sin(t)$$

The system property of interest is the concentration of a fictive substance in the liquid, and in particular we wish to study the dynamic behaviour of the concentration in the two tanks and after the tube. The fact that the tanks are considered well-mixed implies that there is no spatial variation of the concentration in the tanks. We also assume that the flow through the tube is one-dimensional (i.e. it is constant over a cross-section of the tube, so that there is no flow profile due to wall friction etc.), and that it is incompressible (i.e. the flow into the tube is at every moment equal to the flow out of the tube).
The input signal to the process is the concentration in the input flow of the first tank, and the output is the concentration after the tube.

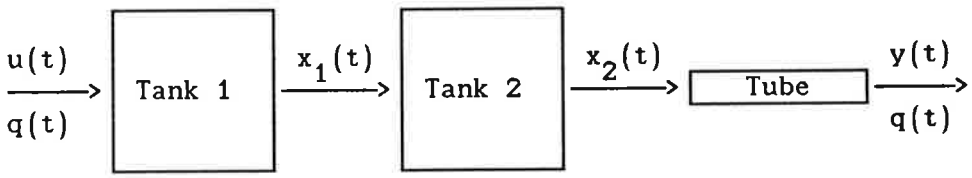


Fig. 2.1 Block diagram of tank and tube system

# 3. PROCESS MODEL

## 3.1 State-space description of the system

In order to formulate a differential equation model for the tanks, we use the condition of mass balance for the dissolved substance in each of the two tanks:

$$\begin{cases} V_1 \dfrac{dx_1}{dt} = -q(t)\cdot x_1(t) + q(t)\cdot u(t) \\ V_2 \dfrac{dx_2}{dt} = q(t)\cdot x_1(t) - q(t)\cdot x_2(t) \end{cases}$$

Notation:

$x_1$ = concentration in tank 1
$x_2$ = concentration in tank 2
$V_1$ = volume of tank 1
$V_2$ = volume of tank 2
$q$ = flow through tanks
$u$ = concentration in input flow of tank 1

The natural choice of state variables for this system is of course the two concentrations, $x_1$ and $x_2$. Since we assume $V_1$ and $V_2$ to be constant we can rewrite the equations above in the standard form:

$$\begin{cases} \dot{x}_1(t) = -\dfrac{q(t)}{V_1} x_1(t) + \dfrac{q(t)}{V_1} u(t) \\ \dot{x}_2(t) = \dfrac{q(t)}{V_2} x_1(t) - \dfrac{q(t)}{V_2} x_2(t) \end{cases} \tag{3.1.1}$$

or, if we use matrix notation:

$$\dot{x}(t) = A(t)\cdot x(t) + B(t)\cdot u(t) \tag{3.1.2}$$

where

$$x(t)=\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \quad ; \quad A(t)=\begin{bmatrix} \dfrac{q(t)}{V_1} & 0 \\ \dfrac{q(t)}{V_2} & \dfrac{-q(t)}{V_2} \end{bmatrix} \quad ; \quad B(t)=\begin{bmatrix} \dfrac{q(t)}{V_1} \\ 0 \end{bmatrix}$$

We note that the resulting model is a second order time variable linear system, and the standard state-space representation can be easily transferred into a

subsystem description in SIMNON.

Considering the tube, we note that the assumption that the flow through the tube is one-dimensional and incompressible permits us to consider the tube as a pure time delay. Unfortunately, a pure time delay cannot be modelled by a finite number of state-space equations, since the concentration state in the tube would have to be modelled by a partial differential equation. It is quite easy, however, to give an adequate input-output description of the tube. This is done in the following section (3.2).

## 3.2 Input-output description of the system

When you make a time domain input-output description of a system, it is very convenient to use the concept of weighting functions. The weighting function h(t,s) for a causal linear system (also known as the impulse response function) is defined by the expression

$$y(t) = \int_0^t h(t,s)\ u(s)\ ds \tag{3.2.1}$$

where

u(t) = input signal to the system
y(t) = output signal from the system when the initial state is zero

If we solve the differential equation for tank 1 (cf. eqn. 3.1.1) with the initial condition $x_1(0) = 0$ we get

$$x_1(t) = \int_0^t \exp\left( -\int_s^t \frac{q(\sigma)}{V_1}\, d\sigma \right) \frac{q(s)}{V_1}\ u(s)\ ds$$

Looking at eqn (3.2.1) we see that $x_1(t)$ can be written as

$$x_1(t) = \int_0^t h_{10}(t,s)\ u(s)\ ds \tag{3.2.2}$$

where

$$h_{10}(t,s) = \exp\left[ -\int_s^t \frac{q(\sigma)}{V_1}\, d\sigma \right] \frac{q(s)}{V_1} \tag{3.2.3}$$

is the weighting function for tank 1.

The differential equation for tank 2 with the initial condition $x_2(0) = 0$ yields

$$x_2(t) = \int_0^t \exp\left( -\int_s^t \frac{q(\sigma)}{V_2}\, d\sigma \right) \frac{q(s)}{V_2}\, x_1(s)\, ds$$

and so $x_2(t)$ can be expressed as

$$x_2(t) = \int_0^t h_{21}(t,s)\, x_1(s)\, ds \tag{3.2.4}$$

where

$$h_{21}(t,s) = \exp\left[ -\int_s^t \frac{q(\sigma)}{V_2}\, d\sigma \right] \frac{q(s)}{V_2} \tag{3.2.5}$$

is the weighting function for tank 2. To calculate the weighting function for both tanks (from u(t) to $x_2(t)$) we insert $x_1(t)$ given by eqn. (3.2.2) into (3.2.4):

$$x_2(t) = \int_0^t h_{21}(t,s) \int_0^s h_{10}(s,\sigma)\, u(\sigma)\, d\sigma\, ds =$$

$$= \int_0^t \int_\sigma^t h_{21}(t,s)\, h_{10}(s,\sigma)\, ds\, u(\sigma)\, d\sigma$$

Again looking at eqn. (3.2.1) we can write

$$x_2(t) = \int_0^t h_{20}(t,s)\, u(s)\, ds \tag{3.2.6}$$

where

$$h_{20}(t,s) = \int_s^t h_{21}(t,\sigma)\, h_{10}(\sigma,s)\, d\sigma \tag{3.2.7}$$

is the weighting function from u(t) to $x_2(t)$.

Finally, to derive the weighting function for the tube we note that we may consider the tube as a pure time delay (cf. section 3.1):

$$y(t) = x_2(t-t_d) \tag{3.2.8}$$

This relation can also be expressed as

$$y(t) = \int_0^t \delta(s-t+t_d)\, x_2(s)\, ds$$

where $t_d$ is the time delay, i.e. the time required for the liquid to pass the tube,

and $\delta(t)$ is the well-known impulsive function, also known as the Dirac (delta-) function. Thus we can identify

$$h_{32}(t,s) = \delta(s-t+t_d) \tag{3.2.9}$$

as the weighting function for the tube. In the general case $t_d$ may of course be a function of time. This will be treated in the next section (3.3).

## 3.3 The variable time delay of the tube

As mentioned earlier (cf. chapter 2) we assume the flow through the tube to be one-dimensional and incompressible. The tube dynamics may therefore be described by the following equation (cf. eqn. 3.2.8):

$$y(t) = x_2(t-t_d(t)) \tag{3.3.1}$$

Here we have stressed the fact that $t_d$ may be a function of time by using the notation $t_d(t)$ for the time delay.

If the flow through the tube is constant, $q(t)=q_0$, then the time delay $t_d$ is also constant and it can be easily calculated as the time required for a particle to pass the tube:

$$t_d = V_3/q_0 \qquad (V_3\text{=tube volume})$$

If the flow $q(t)$ is a function of time, however, the time delay is variable and we have to be more careful. We note that a particle in the tube has a velocity $v(t)$ given by

$$v(t) = q(t)/A \qquad (A\text{=cross-section area})$$

To pass the tube, the particle has to travel a distance $L = V_3/A$. If it starts at $t=t_o$ it reaches the end of the tube at a time $t=t_1$ given by

$$L = \int_{t_o}^{t_1} v(s)\, ds = \int_{t_o}^{t_1} \frac{q(s)}{A}\, ds = \frac{V_3}{A}$$

$$\Rightarrow \qquad V_3 = \int_{t_o}^{t_1} q(s)\, ds$$

This means that the time delay $t_d$ for a particle entering the tube at $t=t_o$ is

$$t_d = t_1 - t_o$$

and we get the following equations for $t_d$:

$$V_3 = \int_{t_o}^{t_o + t_d} q(s)\ ds = \int_{t_1 - t_d}^{t_1} q(s)\ ds \qquad (3.3.2)$$

Now, we are interested in calculating the time delay as a function of the time t when the particle exits from the tube, i.e. $t_d(t)$ (cf. eqn. 3.3.1). Therefore we substitute $t_1$ with t in eqn (3.3.2) and we find that $t_d(t)$ is given by the following implicit expression:

$$V_3 = \int_{t-t_d(t)}^{t} q(s)\ ds \qquad (3.3.3)$$

This equation indicates that not only is $t_d$ a function of time, but it is also generally quite difficult to find an explicit expression for $t_d(t)$. Even a simple sinusoidal flow variation like the one considered,

$$q(t) = q_0 + q_1 \cdot \sin(t)$$

yields a transcendental equation which has to be solved numerically:

$$V_3 = q_0 \cdot t_d(t) + q_1 \cdot (\cos(t - t_d(t)) - \cos(t)) \qquad (3.3.4)$$

How to solve this equation and simulate a variable time delay in SIMNON will be treated in the next chapter.

# 4. SIMULATION OF VARIABLE TIME DELAYS

## 4.1 Solving implicit algebraic equations in SIMNON

The implicit algebraic equation (3.3.4) can be solved by using the optional integration algorithm DAS (Differential Algebraic Systems) in SIMNON. As the name implies, this algorithm has the capability of solving both differential and algebraic equations. DAS has to know which continuous states that are considered slow, fast and algebraic. This partition is done by the command STATE, which has the following syntax:

```
STATE { <state variables> } -<SLOW|FAST|SOLVE>
```

Note that all states are set SLOW by default. SOLVE denotes states defined by implicit algebraic equations. In SIMNON such equations are arranged as follows:

System description

```
        |
    state z
    der  dz
       |
    dz=F(z)
       |
```

Command sequence

```
>algor das          "  choose algorithm DAS
>state z-solve      "  define "algebraic state"
>simu 0 1           "  solve 0=F(z)
```

In this problem, the state z corresponds to the variable time delay $t_d(t)$. The expression F(z) may be described in two principally different ways.

- If the hydraulic flow rate q(t) is known and easy to integrate analytically, then F(z) can be expressed in some elementary functions (cf. eqn. 3.3.3 & 3.3.4). This case will be treated in section 4.2.

- In other cases when F(z) cannot be expressed in elementary functions, another method must be used. This more general but also more complicated method will be treated in section 4.4.

## 4.2 A solution method using DAS

When q(t) can be integrated analytically the problem of calculating the delay $t_d$ is straight-forward. With the previously considered sinusoidal flow rate variation, $q(t)=q_0+q_1 \cdot \sin(t)$, the algebraic equation F(z)=0 is derived by rearranging eqn. (3.3.3) and substituting $t_d(t)$ by the "solve-state" z.

$$0 = \int_{t-z}^{t} q(s)ds - V_3 = q_0 \cdot z + q_1 \cdot \left[\cos(t-z) - \cos(t)\right] - V_3 = F(z)$$

System description

```
continuous system NEWTON
output  td
state    z
der     dz
time     t

dz=q0*z+q1*(cos(t-z)-cos(t))-V3
td=z

q=q0+q1*sin(t)

V3: 1
q0: 1
q1: .5

end
```

Possible command sequence

```
>syst NEWTON
>algor DAS
>state z-solve
>simu 0 10
```

Having found this method for calculating the variable time delay in SIMNON, we can use the standard system DELAY for a straight forward simulation of the transport delay in the tube.

## 4.3 The standard system DELAY

This system stores old values of the variables in vectors. When delayed values are needed, interpolation is performed using Aitken or Hermite algorithms. (If Hermite-interpolation is wanted the derivatives of the variables must also be stored.) A maximum of 10 variables can be delayed.

Global variables are used to specify the choice of interpolation algorithm and the number of elements in the allocation area for saving old values. These variables must be set before the systems are defined by the command SYST.

Fig. 4.1 Block diagram of standard system DELAY

If the previously presented system NEWTON is used to determine the variable time delay $t_d$, a delayed quantity $y=u(t-t_d)$ may be calculated with the DELAY standard system. This can be done by connecting the systems NEWTON and DELAY in a connecting system CPIPE1:

```
connecting system CPIPE1
time t

td1[DELAY]=t-td[NEWTON]
u1[DELAY]=u                  } DELAY inputs

y=y1[DELAY]                    DELAY output

u: 1
end
```

Command sequence

```
>let n1.delay=0                  " no "Hermite variables"
>let n2.delay=1                  " one "Aitken variable"
>let space.delay=500             " allocation area
>syst NEWTON DELAY CPIPE1        " define systems
>algor DAS                       " choose algorithm
>state z-solve                   " define "algebraic state"
>simu 0 10                       " solve and simulate
```

Note that y1[DELAY]=0 when $t-t_d<0$. A non-zero initial y value can be simulated by conditionally assigning this initial value to y in the connecting system while $t<t_d$.

## 4.4 A store-and-search method using DELAY and DAS

When the implicit equation (3.3.3) cannot be expressed in elementary functions, a more general method must be used. The idea is very simple. The implicit equation is an integral equation and it is therefore natural to use the integrated flow rate $Q(t)\equiv\int q(s)ds$. Thus rewrite equation (3.3.3) in terms of Q:

$$V_3 = \int_{t-t_d(t)}^{t} q(s)\,ds = \int_0^t q(s)\,ds - \int_0^{t-t_d(t)} q(s)\,ds =$$

$$= Q(t) - Q(t-t_d(t)) \qquad (4.4.1)$$

Now, if all old Q values are stored in a data vector with corresponding old times, one just has to search in the data vector for an old Q matching the equation. At most one such Q can be found and the corresponding old time is then equal to $t_d$. This is due to the fact that Q is growing monotonically if q(t) is strictly positive.

The standard system DELAY and the algorithm DAS presented earlier are well suited for the task of implementing this store-and-search method. DELAY is used to store old Q and t values and DAS is used for the searching.

The equation (4.4.1) has no solution when $t<t_d \iff Q(t)<V_3$. See fig. 4.2. This problem is taken care of by conditionally solving either the equation z-t=0 or $Q(t)-Q(t-z)-V_3=0$ depending on if $Q(t)<V_3$ or not. The solution of the second equation will be $z=t_d$. Q(t-z) is evaluated using the DELAY system. The DAS solve algorithm must be given a good start value which explains the choice of the equation z-t=0, i.e. $z-t=Q(t)-Q(t-z)-V_3$ when $Q(t)=V_3$.

To summarize and illustrate the method described above the earlier example is presented once more, but this time we do not express the integral of the flow in elementary functions.

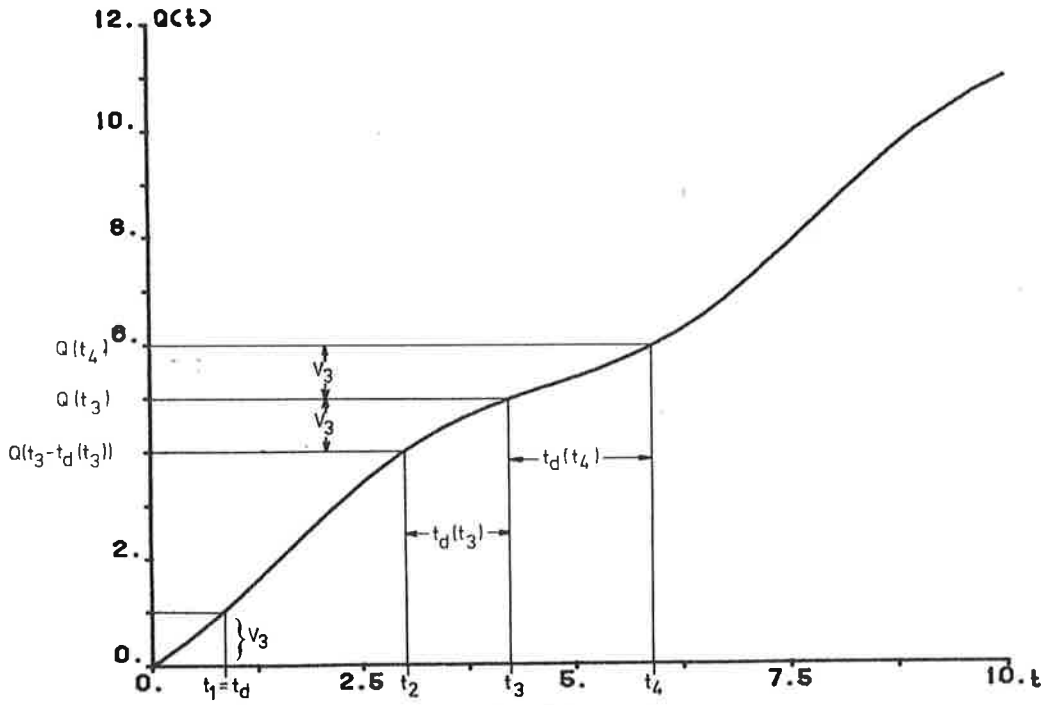


Fig. 4.2 The integrated flow rate $Q(t)\equiv\int^t q(s)ds$.

```
continuous system RAPHSON
input q Qtd
output td Qt
state  intq  z  dum
der   dintq dz ddum
time t

dintq=q
Qt=intq

dz=if Qt<V3 then z-t else Qt-Qtd-V3
td=if Qt<V3 then t else z

ddum= - dum + z


V3: 1

end
```

fictive dynamics keeping the step size down (cf. practical aspects below)

```
connecting system CPIPE2
time t

td1[DELAY]=t-td[RAPHSON]
u1[DELAY]=u
td2[DELAY]=t-td[RAPHSON]
u2[DELAY]=Qt[RAPHSON]

y=y1[DELAY]
Qtd[RAPHSON]=y2[DELAY]

q[RAPHSON]=q0+q1*sin(t)

u: 1
q0: 1
q1: 0.5
end
```

} DELAY inputs

} DELAY outputs

Command sequence

```
>let n1.delay=0                  " no "Hermite variables"
>let n2.delay=2                  " two "Aitken variables"
>let space.delay=500             " allocation area
>syst RAPHSON DELAY CPIPE2       " define systems
>algor DAS                       " choose algorithm
>state z-solve                   " define "algebraic state"
>state dum-fast                  " define "fast dummy state"
>simu 0 10                       " solve and simulate
```

The line "td=if..." in RAPHSON

If the interpolation time t-z is used as input to the DELAY system while $Q(t)<V_3$ problems will occur at the start. The standard system DELAY is then interpolating to time t-z. If this time is negative the corresponding Q(neg. time) is zero. On the other hand if t-z is positive the corresponding Q(pos. time) must be interpolated. At the first step the DELAY system has too few values stored and an interpolation can not be performed. Now, there is no need for the DELAY system when $t<t_d$. Thus, set the interpolation time input to zero (td1[DELAY]=0) when $t<t_d$. This could be arranged in the connecting system but it is perhaps nicer to use a conditional $t_d$ as follows:

```
continuous system RAPHSON
                 |
td=if Q<V3 then t else z
                 |
```

```
connecting system CPIPE2
time t
td1[DELAY]=t-td[RAPHSON]
                 |
```

Some practical aspects

Because of the fact that the algebraic equations are time dependent, too long step sizes will provide the DAS solve algorithm with bad start values. If there are no more dynamics in the system than described so far, the step size will increase causing worse and worse start values for the solving iterations. When simulating realistic systems, you normally have more dynamics and several states. Thus the step size will be kept down automatically. To avoid this problem here, where the ideas are illustrated in a simplified example, a dummy fast state variable (dum) has been introduced to keep the step size down.

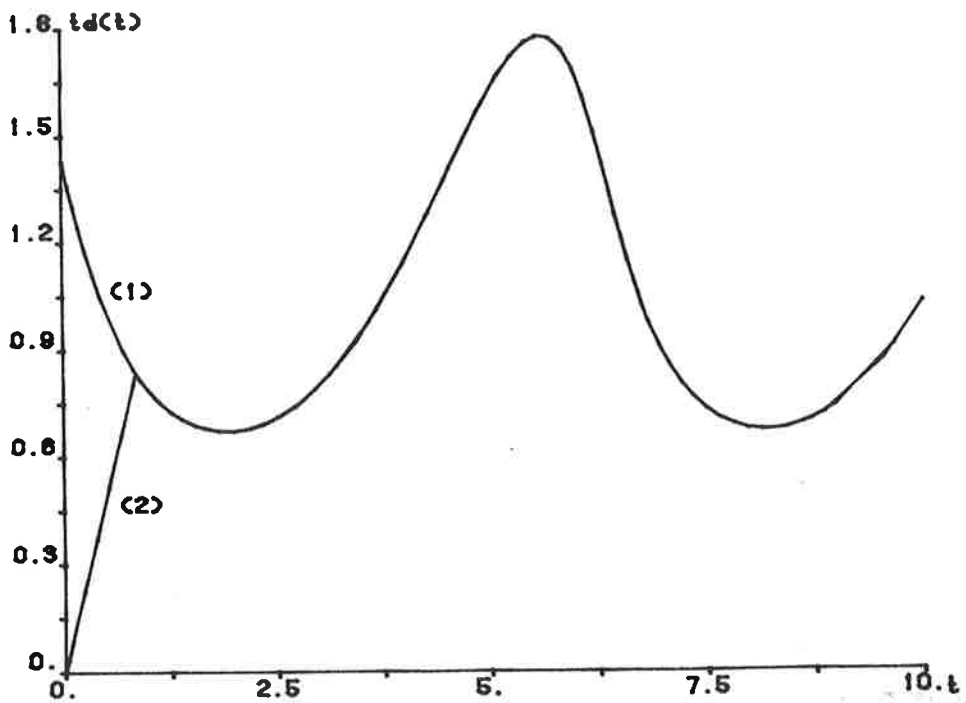


Fig. 4.3 The variable time delay $t_d(t)$ calculated with the methods described in 4.2 (curve 1) and 4.4 (curve 2).

# 5. SIMULATION OF CONCENTRATION DYNAMICS

## 5.1 General

We now return to the task of simulating the concentration dynamics of the tank process model described in chapter 3. In order to give the system description a well-organized structure in SIMNON, we used five subsystems in the simulations.

(1) Continuous system TANKS
This system contains the dynamics of the two tanks (cf. section 3.1) and was used for simulating the step response and the adjoint equations described in section 5.2 and 5.3 below.

(2) Continuous system FLOW
This system was used for calculating the time-variable flow q(t) and the tube time delay $t_d$ with the method described in chapter 4.

(3) Standard system DELAY
Simulates the time delay of the tube.

(4) Connecting system CONTANK
In CONTANK, the output from the subsystem TANKS (=state $x_2$) is connected to the input of the tube (standard system DELAY). The output signal from DELAY corresponds to the output of the tube y(t). CONTANK also feeds DELAY with the current value of $t_d$ from subsystem FLOW, and supplies the subsystem TANKS with the input concentration u(t).

(5) Continuous system GRAM
This system contains the dynamics of the tanks and also the necessary equations for the simulation of Gramians and controlling the system to zero, as described in section 5.4 and 5.5 below.

The systems TANKS, FLOW, DELAY and CONTANK form a unit and must be compiled together, whereas GRAM is an independent system and should be compiled and used separately. A listing of the SIMNON source code for the subsystems TANKS, FLOW, CONTANK and GRAM is given in appendix A. DELAY is a standard subsystem in SIMNON and it is not available for listing. In appendix A you will also find a macro, INITANK. It was used to compile the subsystems TANKS, FLOW, CONTANK and DELAY and to initialize DELAY and the DAS integration algorithm before the simulation.

All simulations have been made using the following values of the system parameters:

$V_1$ = 3.0 (volume of tank 1)
$V_2$ = 2.0 (volume of tank 2)
$V_3$ = 1.0 (volume of tube)
$q_0$ = 1.0 (average flow value)
$q_1$ = 0.5 (amplitude of flow oscillations)

Unless otherwise stated, all simulations have been made from the initial condition $x_1(0) = x_2(0) = y(0) = 0$ .

## 5.2 Tank system step response

The tank system step response was simulated using the subsystems TANKS, FLOW, DELAY and CONTANK and the following input concentration signal:

$$u(t) = \begin{cases} 0.1 & ; \ t>0 \\ 0 & ; \ t<0 \end{cases}$$

The simulation result, a plot of the step response in $x_1(t)$, $x_2(t)$ and $y(t)$ is shown in fig. 1, appendix B. Note that the step response at another time may look different since the tank system is time variable.

## 5.3 Adjoint differential equations

The concept of the adjoint equation associated with a given equation is described in chapter 7 of Brockett [1]. According to theorem 7.1, the adjoint equation associated with the homogeneous equation

$$\dot{x}(t) = A(t)x(t)$$

is

$$\dot{p}(t) = -A^T(t)p(t) \tag{5.3.1}$$

This equation was included in the subsystem TANKS using the A matrix defined in eqn. (3.1.2) and the state vector

$$p(t) = \begin{bmatrix} p_1(t) \\ p_2(t) \end{bmatrix}$$

The inner product

$$\langle x(t), p(t) \rangle = x_1(t)p_1(t) + x_2(t)p_2(t)$$

(which is constant according to the definition of adjoint equations) was also calculated in TANKS.

In order to simulate the homogeneous equations the input signal u(t) was set to zero during the simulation. The initial conditions for the simulation were:

$$\begin{cases} x_1(0) = x_2(0) = 0.5 \\ p_1(0) = p_2(0) = 0.1 \end{cases}$$

The results of the simulation from t=0 to t=12 are shown in fig. 2 and 3 in appendix B.

## 5.4 Reachability and controllability Gramians

The reachability Gramian for the system

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \qquad ; \quad x(t_0) = x_0$$

is defined as

$$W_r(t_0,t) = \int_{t_0}^{t} \Phi(t,s)B(s)B^T(s)\Phi^T(t,s)\, ds$$

where $\Phi(t,t_0)$ is the transition matrix for the system.

$W_r$ can be used to determine which states that can be reached from the origin ($x(t_0) = 0$), since all reachable states can be shown to lie in the range space of $W_r$. It is easy to verify that $W_r$ is symmetric and that it satisfies the differential equation

$$\begin{cases} \frac{d}{dt} W_r(t_0,t) = A(t)W_r(t_0,t)+W_r(t_0,t)A^T(t)+B(t)B^T(t) \\ W_r(t_0,t_0) = 0 \end{cases} \tag{5.4.1}$$

In our case, with $W_r$ being a symmetric 2x2 matrix, eqn. (5.4.1) above yields three scalar differential equations for the elements of $W_r$. These were used to simulate $W_r$.

The controllability Gramian, $W_c$, can be used to determine from which states $x(t_0)$ the system can be controlled to the origin ($x(t_1) = 0$), and also to calculate a control signal to accomplish this transfer (this will be treated in the next section, (5.5)). When the reachability Gramian $W_r$ is known, the controllability Gramian can be calculated from the expression

$$W_c(t_0,t) = \Phi(t_0,t)W_r(t_0,t)\Phi^T(t_0,t) \tag{5.4.2}$$

where

$$\Phi(t_0,t) = \Phi^{-1}(t,t_0) \tag{5.4.3}$$

Therefore we first calculate the transition matrix $\Phi(t,t_0)$, and then $W_c$ can be calculated during the simulation using eqn. (5.4.2) and (5.4.3).
Since $\Phi(t,t_0)$ satisfies the homogeneous equation

$$\dot{\Phi}(t,t_0) = A(t)\Phi(t,t_0) \quad ; \quad \Phi(t_0,t_0) = I \tag{5.4.4}$$

it is easy to simulate $\Phi(t,t_0)$ by solving the differential equation corresponding to eqn (5.4.4) for each matrix element. In this particular case, with the A matrix defined as in eqn (3.1.2), we immediately get the result that element (1,2) of $\Phi$ is zero for all values of t. The other three elements are simulated with eqn (5.4.4).

It would perhaps be tempting to simulate the inverse transition matrix $\Phi(t_0,t)$ directly (it satisfies a differential equation corresponding to eqn. (5.4.4), cf. Brockett [1], chapter 7), but when you have a stable A matrix as in our case, this results in an unstable equation and for numerical reasons it is not to be recommended.

We have also simulated the condition number for $W_r$ and $W_c$. The condition number cond(M) for a matrix M indicates how sensitive the solution to the equation

$$y = Mx$$

is to uncertainties in the elements of y and M. A large condition number indicates a badly conditioned problem (very sensitive to element uncertainties). For symmetric matrices, the condition number can be calculated as the ratio between the largest and the smallest eigenvalue of the matrix:

$$\text{cond}(M) = \frac{\lambda_{max}(M)}{\lambda_{min}(M)} \tag{5.4.5}$$

Since $W_r$ and $W_c$ are 2x2 matrices, the eigenvalues can be calculated by solving their second order characteristic equations. Both matrices being symmetric, we also know that the eigenvalues are real and consequently we do not have to worry about complex roots to the characteristic equations.

The simulation results, showing plots of the elements in $W_r$ and $W_c$ as well as the corresponding condition numbers are presented in fig. 4-7 in appendix B.

## 5.5 Controlling the system from an initial state to zero

The problem of controlling a system from an arbitrary initial state $x(t_0) = x_0$ to the origin in a finite time $t_1$ is treated in chapter 13 of Brockett [1]. Theorem 13.1 says that if $x_0$ belongs to the range space of $W_c$ then a control signal which accomplishes the desired transfer can be calculated from

$$u(t) = -B^T(t)\Phi^T(t_0,t)\ \eta \tag{5.5.1}$$

where $\eta$ is any solution to

$$W_c(t_0,t_1)\ \eta = x_0 \tag{5.5.2}$$

To simulate the transfer from $x_0$ to $x(t_1) = 0$, it is necessary to proceed in two steps:

1) First simulate the system from the initial state $x(0) = x_0$ to the desired time $t=t_1$ without any input signal ($u(t) = 0$). During this simulation $\eta$ is calculated "on line" by solving eqn. (5.5.2). At the end of the simulation, when $t = t_1$, you have a vector $\eta$ which can be used in the control law (5.5.1).

2) Then repeat the simulation from $t=0$ to $t=t_1$ with the same initial condition as in step 1, but with the input signal given by eqn (5.5.1) and using the $\eta$-vector calculated in step 1.

A SIMNON macro called ZERO for this two step procedure is given in appendix A. You may note the convenient transfer of the calculated elements in the $\eta$-vector from the first to the second simulation.

The procedure was tested by trying two different values for the final time $t_1$ ($t_1=6$ and $t_1=12$). The initial conditions in both cases were

$$x_1(0) = 0.5$$
$$x_2(0) = 0.5$$

The results of the simulations are presented in fig. 8 and 9 in appendix B.

You may note when looking at the results that the input signal required to bring the system from $x(t_0) = x_0$ to $x(t_1) = 0$ is much smaller for $t_1 = 12$ than for $t_1 = 6$. This is quite natural, since the system is stable. The states will therefore approach zero asymptotically even if no control signal is present. Of course the need for external influences on the system (control signals) is smaller if we allow a larger time interval for the transfer to the origin.

## 6. REFERENCES

[1] Brockett, Roger W.: "Finite Dimensional Linear Systems". John Wiley & Sons Inc., 1970.

[2] Elmqvist, H.: "SIMNON - User's Manual". Dept of Automatic Control, Lund Institute of Technology. Report no 7502, April 1975.

[3] Åström, Karl Johan: "A SIMNON Tutorial". Dept of Automatic Control, Lund Institute of Technology. LUTFD2/(TFRT-3168)/1-52/(1982).

```
CONTINUOUS SYSTEM TANKS

" Concentration dynamics of two cascaded, well-mixed tanks
" Simulation of step response and adjoint equations

"  u  = input concentration
"  q  = flow through tanks
"  v1 = volume of tank 1 (assumed constant)
"  v2 = volume of tank 2 (     - " -      )
"  x1 = concentration in tank 1
"  x2 = concentration in tank 2

"  p1 = state 1 in adjoint tank equation
"  p2 = state 2 in adjoint tank equation
"  xtp = standard inner product  <x(t),p(t)>

" Note: u must be zero when simulating adjoint equations

INPUT   u    q
STATE   x1  x2  p1   p2
DER    dx1 dx2 dp1  dp2
TIME   t

a = q/v1
b = q/v2

dx1 = -a*x1       + a*u
dx2 =  b*x1 -b*x2

dp1 =  a*p1 -b*p2
dp2 =       b*p2

xtp = x1*p1 + x2*p2

v1 : 3.0
v2 : 2.0

END
```

```
CONTINUOUS SYSTEM FLOW

" Calculation of time-variable flow and tube delay

" q  = tank flow
" v3 = volume of delay tube
" z  = variable time delay ( z=f(q,t))

OUTPUT  q
STATE   z
DER     dz
TIME    t

q = q0 + q1*sin(t)

dz = q0*z + q1*(cos(t-z) - cos(t)) - v3

q0 : 1.0
q1 : 0.5
v3 : 1.0

END
```

```
CONNECTING SYSTEM CONTANK

" Generates input signal to subsystem TANKS and connects
" the output of TANKS (=state x2) to the input of the
" tube (simulated with standard system DELAY)

"   u  = input concentration to tanks
"   q  = flow through tanks and tube (calculated in subsystem FLOW)
"   z  = variable time delay in tube (          - " -          )
"   y  = output from delay tube

TIME t

u[tanks] = u0
q[tanks] = q[flow]
td1[delay] = t - z[flow]
u1[delay] = X2[tanks]
y = y1[delay]

u0 : 0.1

END
```

```
CONTINUOUS SYSTEM GRAM

" Concentration dynamics of two cascaded, well-mixed tanks
" Simulation of Gramians and controlling the system to zero

"  u  = input concentration
"  n1,n2 = elements in vector defining input signal to drive x(t) to zero
"  q  = flow through tanks
"  v1 = volume of tank 1 (assumed constant)
"  v2 = volume of tank 2 (      - " -       )
"  x1 = concentration in tank 1
"  x2 = concentration in tank 2

"  f11 = element (1,1) of transition matrix Fi(t,to)
"  f21 =   -"-   (2,1)        - " -
"  f22 =   -"-   (2,2)        - " -
"  g11 =   -"-   (1,1)        - " -          Fi(to,t)
"  g21 =   -"-   (2,1)        - " -
"  g22 =   -"-   (2,2)        - " -
"  wr1 = element (1,1) of reachability Gramian (Wr)
"  wr2 =   -"-   (1,2) of     - " -        -"-
"  wr3 =   -"-   (2,2) of     -"-          -"-
"  ewrmax = maximum eigenvalue of Wr
"  ewrmin = minimum eigenvalue of Wr
"  condwr = cond(Wr)
"  wc1 = element (1,1) of controllability Gramian (Wc)
"  wc2 =   -"-   (1,2) of     - " -        -"-
"  wc3 =   -"-   (2,2) of     -"-          -"-
"  ewcmax = maximum eigenvalue of Wc
"  ewcmin = minimum eigenvalue of Wc
"  condwc = cond(Wc)

STATE   x1  x2  wr1  wr2  wr3  f11  f21  f22
DER     dx1 dx2 dwr1 dwr2 dwr3 df11 df21 df22
TIME    t
INITIAL
   x1=x01
   x2=x02
SORT

q = q0 + q1*sin(t)
a = q/v1
b = q/v2
u = -a*(g11*n1+g21*n2)

dx1 = -a*x1          + a*u
dx2 =  b*x1  -b*x2

df11 = -a*f11
df21 =  b*f11 -b*f21
df22 =               -b*f22

dwr1 = -2*a*wr1                    + a*a
dwr2 =    b*wr1 -(a+b)*wr2
dwr3 =            2*b*wr2 - 2*b*wr3
```

```
g11 = 1.0/f11
g21 = -f21/(f11*f22)
g22 = 1.0/f22

wc1 = wr1*g11*g11
wc2 = g11*(wr1*g21+wr2*g22)
wc3 = g21*(wr1*g21+wr2*g22) + g22*(wr2*g21+wr3*g22)

swr = sqrt((wr1-wr3)*(wr1-wr3)/4.0 + (wr2*wr2))
ewrmax = (wr1+wr3)/2.0 + swr
ewrmin = (wr1+wr3)/2.0 - swr
condwr = if (abs(ewrmin)<1.0E-6) then 1.0E6 else ewrmax/ewrmin

swc = sqrt((wc1-wc3)*(wc1-wc3)/4.0 + (wc2*wc2))
ewcmax = (wc1+wc3)/2.0 + swc
ewcmin = (wc1+wc3)/2.0 - swc
condwc = if (abs(ewcmin)<1.0E-6) then 1.0E6 else ewcmax/ewcmin

neta = (wc1*wc3-wc2*wc2)
eta1 = if abs(neta)>1e-3 then (wc3*x01-wc2*x02)/neta else 1e3
eta2 = if abs(neta)>1e-3 then (wc1*x02-wc2*x01)/neta else 1e3

" initial system parameter values
q0 : 1.0
q1 : 0.5
v1 : 3.0
v2 : 2.0
n1 : 0.0
n2 : 0.0

" initial state values
x01 : 0.0
x02 : 0.0
f11 : 1.0
f21 : 0.0
f22 : 1.0

END
```

```
MACRO INITANK

"  Set global variables for standard system DELAY
   LET n1.delay = 0
       ,n2.delay = 1
       ,space.delay = 500

   SYST TANKS FLOW DELAY CONTANK

"  Select integration algorithm
   ALGOR DAS

"  Define the state z as the solution to
"  an implicit algebraic equation
   STATE z[flow] - SOLVE

END
```

```
MACRO ZERO T

" Two step procedure for simulation of the tank system
" controlled from the initial state x1(0)=0.5, x2(0)=0.5
" to the final state x1(T)=0, x2(T)=0.

   PAR x01:0.5  " initiate state vector
   PAR x02:0.5

   PAR n1 :0.0  " reset input signal to zero
   PAR n2 :0.0

   PLOT         " inhibit plot during step 1 simulation
   SIMU 0 T

   PAR n1:eta1  " assign calculated eta1 and eta2 values
   PAR n2:eta2  " to control law parameters

"  Repeat simulation and plot result
   PLOT x1 x2 u
   AXES h 0 T v -0.5 0.5
   SIMU 0 T

END
```

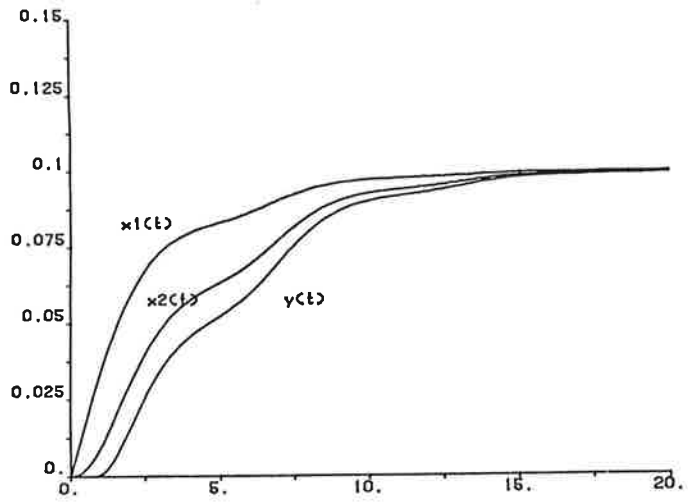


Fig. 1 Tank system step response.

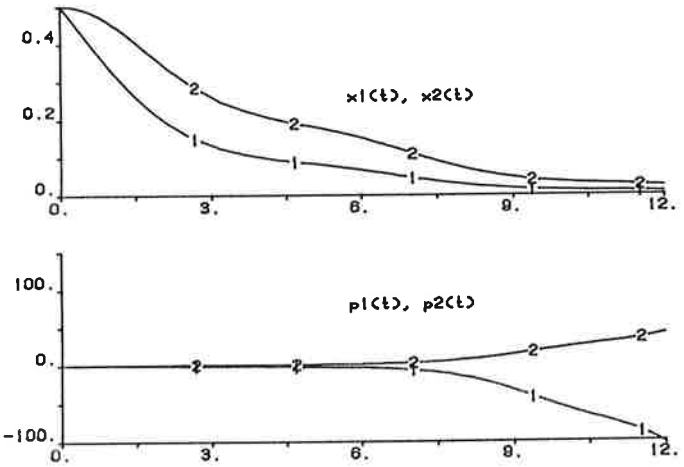


Fig. 2 Simulation of adjoint equations: x(t) and p(t)

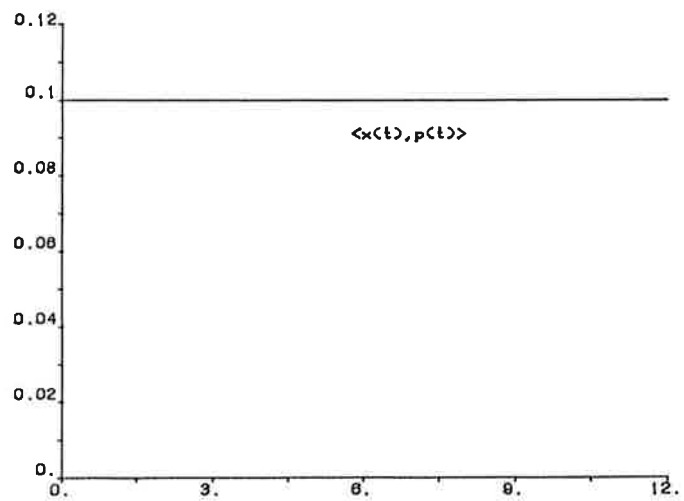


Fig. 3 Simulation of adjoint equations: <x(t),p(t)>

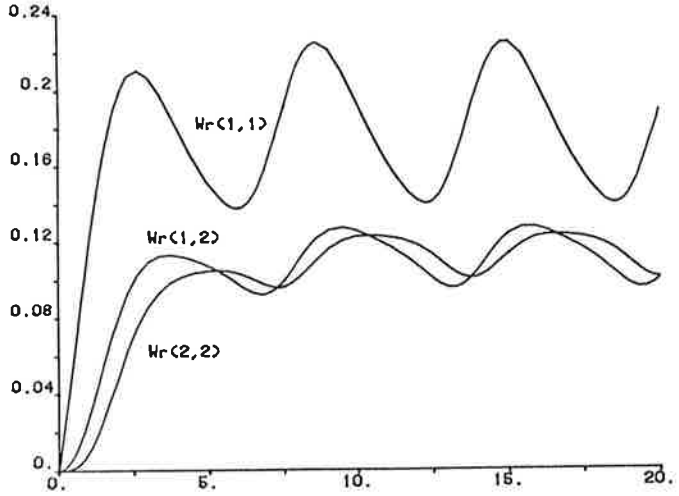


Fig. 4 Simulation of reachability Gramian Wr

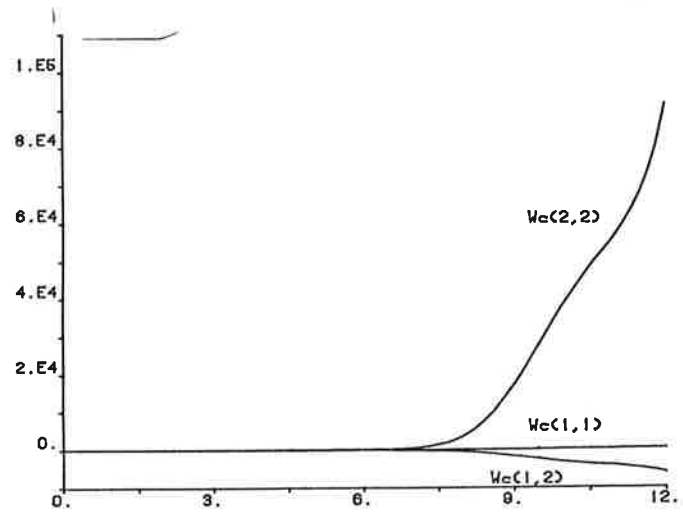


Fig. 5 Simulation of controllability Gramian Wc

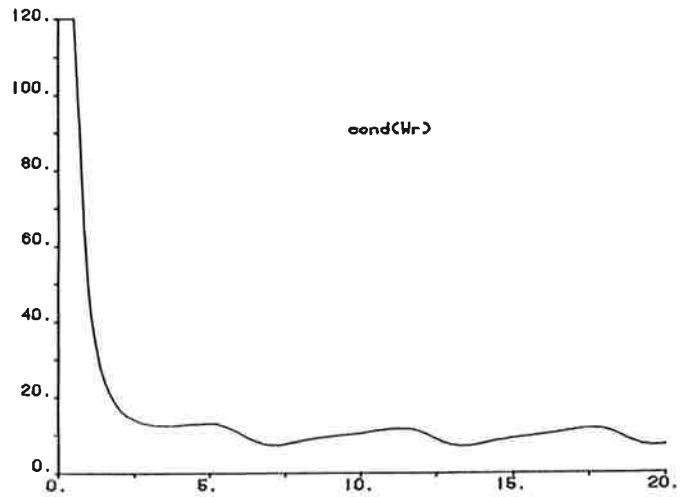


Fig. 6 Simulation of condition number for Wr

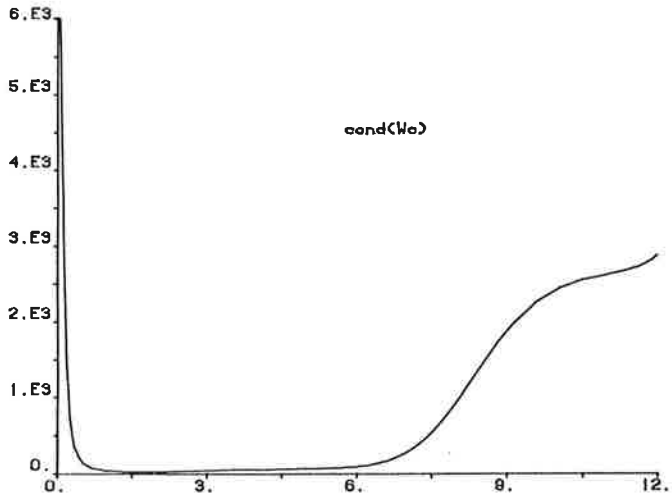


Fig. 7 Simulation of condition number for Wc

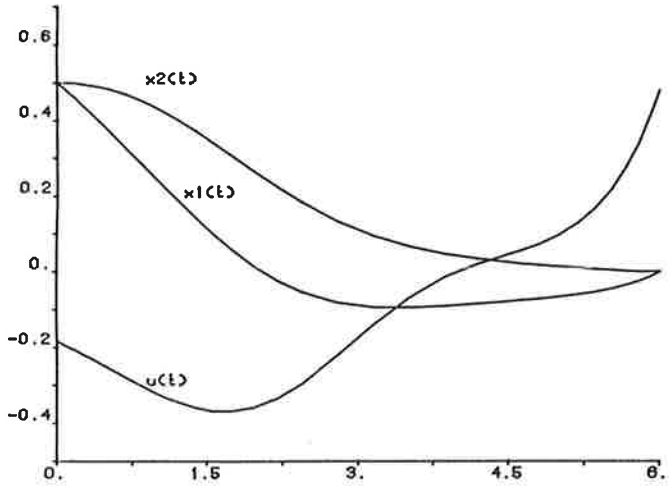


Fig. 8 Controlling the system to zero at t=6

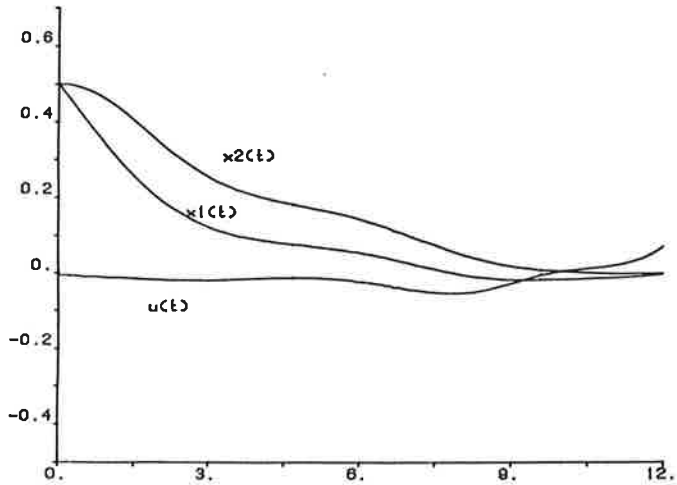


Fig. 9 Controlling the system to zero at t=12