CHAPTER 7

SIMULATION OF SOFTWARE PROJECTS

"The world is teeming with problems. Whenever man looks he encounters some new problem - in his home life and in his job, in economics and in technology, in the arts and in the sciences. And some problems are very stubborn - they refuse to leave us in peace. They torture our thoughts, sometimes haunting us throughout the day and even robbing us of sleep at night."

Max Planck
7.1 **Simulation**

Sometimes the behaviour of a mathematical model of a system may be studied by analytical methods which provide general solutions. When this is not possible numerical computation techniques come to the rescue. The term 'simulation' is used to describe any procedure of establishing a dynamic model and deriving a solution numerically - solutions in steps which may be repeated for various ranges.

Software development process is a continuous system because the predominant activities of the system cause smooth changes in the attributes of the system entities. When this system is modelled mathematically, the variables of the model representing the attributes are controlled by continuous functions. Since the equation (6.3) has non-linearity we apply simulation method to this continuous model.

7.2 **Continuous System Modeling Program**

In order to simulate continuous system on digital computers a number of continuous system simulation languages (CSSLs) have been developed. They use the familiar statement-type of input, allowing a problem to be programmed directly from the equations of a mathematical model, rather than
requiring the equations to be broken into functional elements.

One of the popular CSSLs is CSMP language (To 79). As an illustration, a program in CSMP for software development process is shown in Fig. 7.2. Fig. 7.1 shows the functional block diagram. Fig 7.3 shows the solution.

```plaintext
TITLE SOFTWARE SYSTEM SIMULATION

PARAM TD = (12.,11.,10.,9.,8.,7.,6.)

Y2DOT = K*F - K*Y - E*YDOT
YDOT = INTGRL (0.0, Y2DOT)
Y   = INTGRL (0.0, YDOT)
E   = K*TIME
K   = 1.0/(TD*TD)

CONST F = 60.0

TIMER DELT = 1 , FINTIM = 3*12 , PRDEL = 1 , OUTDEL=1
PRINT Y, YDOT, Y2DOT
PRTPLT YDOT
LABEL MANPOWER VERSUS TIME
END
STOP

Fig. 7.2 - CSMP III Program of Software System
```
7.3 **Feedback Systems**

For all subcycles of software process - effort expended is proportional to the output (deliverables). A significant factor in the performance of software project is that a coupling occurs between the input (manpower, effort, resources etc.) and output (documentation, code etc.). The term 'feedback' is used to describe this phenomenon (as in fridge, autopilot, satellite etc.). Many simulation studies of continuous systems are concerned primarily with the study of servomechanisms, the general name given to devices that rely upon feedback for their operation. The field of control theory provides the theoretical background with which to design such systems and simulation is extensively used to carry out detailed studies. For example - simulation of aircraft under control of autopilot.

7.4 **Auto-controller for Software System**

The error signal $\phi^o$ is defined as the difference between the desired heading or input effort $\phi_i$ and the actual heading or output $\phi_0$ -

$$\text{Error } \Delta \phi = \phi_i - \phi_0 \quad \ldots \ldots \quad (7.1)$$

A workforce is required to change the project heading which is proportional to $\Delta \phi$. This turning force $F$ produces entropy (resistance or drag) which is proportional to effort rate or manpower $\phi'$ -

$$\text{Turning force } F = \text{workforce - entropy loss}$$

$$= k(\Delta \phi) - \epsilon \phi_0 \quad \ldots \ldots \quad (7.2)$$
The fundamental law of work mechanics is that 'pace' (acceleration) of a project is proportional to the applied force

\[ \Phi_O = F \quad \ldots \ldots \quad (7.3) \]

Substituting from (7.1) and (7.2) and transposing terms the resultant equation is

\[ \Phi_O + \epsilon \Phi_O + \delta \Omega_o = \delta \Phi_i \]

Assuming \( R = 1 \) unit, \( \delta = w^2 \) and \( \epsilon = 2 \zeta w \) the above equation may be transformed to the following form -

\[ \Phi_o + 2 \zeta w \Phi_o + w^2 \Phi_o = w^2 \Phi_i \quad \ldots \ldots \quad (7.4) \]

This is II order differential equation and describes the relationship between output and input for the software system.

Suppose a project is progressing at a steady course which, by definition, we take to be zero heading. If it is asked to change to a new heading at time \( t \), this corresponds to a change of input (requirements/specs change). The project will settle to a new heading as shown in Fig. 7.5. The response will be stabilized if \( \zeta > 1 \).

7.5 CSMP Simulation of Auto-controller

In order to simulate how auto-controller can be designed to modify the project progress, it is more convenient to treat the model in the form of individual equations. Using the variables they may be written as follows:

\[
\begin{align*}
\text{ERROR} &= \Phi_i - \Phi_o \\
\text{TFORCE} &= \delta \times \text{ERROR} - \epsilon \times \text{MPOWER} \\
R \times \text{PACE} &= \text{TFORCE}
\end{align*}
\]
Further -

\[
\begin{align*}
\text{ERROR} & = \text{INPUT} - \text{HEAD} \\
\text{TFORCE} & = \text{ALPHA} \times \text{ERROR} - \text{E} \times \text{MPOWER} \\
\text{RS} \times \text{PACE} & = \text{TFORCE}
\end{align*}
\]

A CSMP III program for the auto-controller is shown in Fig. 7.5.

**TITLE**

* 

**PARAM**

\[E = (12, 18, 24, 30)\]

\[
\begin{align*}
\text{INPUT} & = 1.2 \times \text{HEAD} \\
\text{ERROR} & = \text{INPUT} - \text{HEAD} \\
\text{TFORCE} & = \text{ALPHA} \times \text{ERROR} - \text{E} \times \text{MPOWER} \\
\text{PACE} & = \text{TFORCE} / \text{RS} \\
\text{HEAD} & = \text{INTGRL}(0.0, \text{MPOWER}) \\
\text{MPOWER} & = \text{INTGRL}(0.0, \text{PACE}) \\
\text{E} & = \text{ALPHA} \times \text{TIME} \\
\text{ALPHA} & = 1.0 / (\text{TD} \times \text{TD})
\end{align*}
\]

* 

**CONST**

\[\text{RS} = 1.0\]

**TIMER**

\[\text{DELT} = 1, \text{FINTIM} = 4.0 \times 12, \text{PRDEL} = 1, \text{OUTDEL} = 1\]

**PRNT**

\[\text{HEAD}, \text{INPUT}\]

**LABEL**

\[\text{HEADING VERSUS TIME}\]

**END**

**STOP**

---

Fig. 7.4 - CSMP III Program for Simulating Auto-controller
The results of the runs are plotted in Fig. 7.5. They are in non-dimensional form by plotting $w\phi_0/a$ against $wt$. The ILC curve through the origin is the desired heading, which would be followed by the project if it responded perfectly.

7.6 **Interactive and Real-time Simulation**

Software project being a continuous system, its simulation is concerned with its response over time. Visual observation of the results, as they develop, is very important. Most CSSLs provide the ability to display the outputs on plotter or VDU terminal screen. If this capability is provided a manager can see whether the output developing is satisfactory or not. The project profile can be adjusted by such simulation by manipulating development time and manpower. If the lightpen facility is there, the new trajectory can be drawn and the new parameters deducted.

Similarly on these lines, a real-time simulator for software projects could be designed. If there are requirements / specs or manpower changes at some stage, the signal is sent in and the device displays the new trajectory. This is very much useful in project execution and monitoring because it visually displays in which direction the project is progressing against the background of estimates in the beginning of the project.
FIG. 7.1—FUNCTIONAL BLOCK DIAGRAM OF SOFTWARE SYSTEM

FIG. 7.3—SOFTWARE SYSTEM SIMULATION
FIG. 7.5 - AUTO-CONTROLLER FOR SOFTWARE SYSTEM