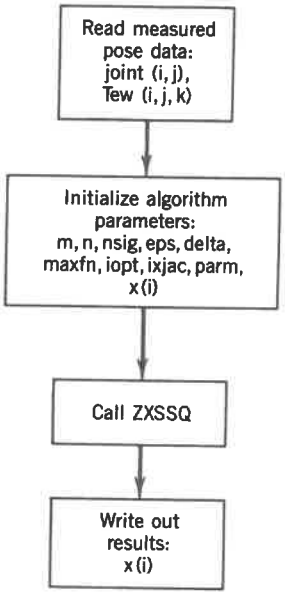

Figure 6.13. Main program block.

initialized to zero. The routine ZXSSQ is called next. On return from ZXSSQ, the error code and sum of the squared errors are printed along with the updated model parameters.

The IMSL subroutine ZXSSQ actually accomplishes the identification. The argument list for the subroutine is as follows:

Call ZXSSQ(Func, m, n, nsig, eps, delta, maxfn, iopt, parm,

x, ssq, f, xjac, ixjac, xjtj, work, infer, ier)

The first parameter, **Func**, is simply the name of the subroutine that calculates the error values. In this example, the name **PIndex** was used. The argument **m** and **n** are the number of equations and the number of unknown model coefficients. Since we measured 50 complete poses with six unique parameters (three position and three orientation) per pose, **m** has a value of 300. As mentioned above, each model used has 30 coefficients, which is the value of **n**. The parameter **nsig** is the number of significant digits desired in the model parameters. In this example, **nsig** is set to 4 and is the only convergence criterion that is used. If, on two successive iterations, the coefficient estimations agree component by component to four digits, the process is deemed to have converged and is stopped. The arguments **eps** and **delta** are additional convergence criteria that are set to 0 and not used here. A more complete description of these arguments is given in Appendix A. The parameter **maxfn** is the maximum number of times the subroutine **Func (PIndex)** may be called. Since we wish to give the algorithm ample

APPENDIX C

SUBROUTINES ASSOCIATED WITH ZERO REFERENCE POSITION METHOD

```
Subroutine Par(x, Param, Offset )
Real*8 x(30), Param(7,6), Offset(6)

Param(1,1) = x(1)
Param(1,2) = (1.0d0 -x(1)**2 - x(2)**2)**.5
Param(1,3) = x(2)

Param(1,4) = x(3) - 15.0d0
Param(1,5) = 14.70d0
Param(1,6) = x(4) + 33.60d0

Param(2,1) = x(5)
Param(2,2) = x(6)
Param(2,3) = -(1.0d0 -x(5)**2 - x(6)**2)**.5

Param(2,4) = x(7) - 15.0d0
Param(2,5) = x(8) + 14.70d0
Param(2,6) = 27.730d0

Param(3,1) = x(9)
Param(3,2) = x(10)
Param(3,3) = -(1.0d0 -x(9)**2 - x(10)**2)**.5

Param(3,4) = x(11) + 2.00d0
Param(3,5) = x(12) + 14.70d0
Param(3,6) = 27.730d0

Param(4,1) = x(13)
Param(4,2) = (1.0d0 -x(13)**2 - x(14)**2)**.5
Param(4,3) = x(14)

Param(4,4) = x(15) + 1.20d0
Param(4,5) = 31.747d0
Param(4,6) = x(16) + 27.730d0

Param(5,1) = x(17)
Param(5,2) = x(18)
Param(5,3) = -(1.0d0 -x(17)**2 - x(18)**2)**.5
```

```
      Param(5,4) = x(19) + 1.20d0
      Param(5,5) = x(20) + 31.747d0
      Param(5,6) = 27.730d0

      Param(6,1) = x(21)
      Param(6,2) = (1.0d0 -x(21)**2 - x(22)**2)**.5
      Param(6,3) = x(22)

      Param(6,4) = x(23) + 1.20d0
      Param(6,5) = 31.747d0
      Param(6,6) = x(24) + 27.730d0

      Do 1 i = 1, 6
         Offset(i) = x(24+i)
      Continue

      Return
      End

      Subroutine Forward( Param, Theta, D )
      Implicit Real*8 (a-h,o-z)
      Real*8 D(4,4), Theta(6), Param(7,6)
      Real*8 T1(4,4), T2(4,4), u(3), p(3)

      D(1,1) =  0.0d0
      D(1,2) = -1.0d0
      D(1,3) =  0.0d0
      D(1,4) =  1.20d0
      D(2,1) =  0.0d0
      D(2,2) =  0.0d0
      D(2,3) =  1.0d0
      D(2,4) =  33.960d0
      D(3,1) = -1.0d0
      D(3,2) =  0.0d0
      D(3,3) =  0.0d0
      D(3,4) =  27.730d0
      D(4,1) =  0.0d0
      D(4,2) =  0.0d0
      D(4,3) =  0.0d0
      D(4,4) =  1.0d0

      Do 1 icnt = 1, 6
         i = 7 - icnt

         u(1) = Param(i,1)
         u(2) = Param(i,2)
         u(3) = Param(i,3)

         p(1) = Param(i,4)
         p(2) = Param(i,5)
         p(3) = Param(i,6)

         th   = Theta(i)

         Call Vecrot(u, p, th, T1)

         Call Mamult(T1, D, T2)

         Do 2 ii= 1, 4
            Do 2 jj = 1, 4
               D(ii,jj) = T2(ii,jj)
2        Continue

1     Continue

      Return
      End
```