维普资讯 http://www.cqvip.com

【产品开发与设计】

基于 Simulink/S-Function 的惯性导航系统的仿真*

赵 芳,李永红

(中北大学 信息与通信工程学院,太原 030051)

摘要:利用 Simulink 和 C 语言编写的 S-函数对惯性导航系统进行建模和仿真,并通过 Matlab 命令窗口进行了验证,仿真和试验结果一致度比较高,证明了该方法的正确性和可行性.

关键词:S-Function;惯性导航系统;建模;仿真

中图分类号:V24 **文献标识码**:A **文章编号**:1006-0707(2008)01-0086-04

惯性导航技术是集现代数学、现代控制理论与计算机技术于一体的一项综合性新技术,其算法比较复杂,矩阵计算较多. 因此使用一般的编程软件实现起来比较困难,同时,在惯性导航领域中,对低成本的要求日趋明显和迫切,但对试验的要求却越来越严格和苛刻,高昂的试验费用,繁琐的试验过程,费时费力,如果只是算法或其它非试验方法本身的问题而导致重新试验,则代价有些太大. 因此,本研究考虑了使用 Simulink 来进行仿真,以避免上述问题的出现.

1 用 C 语言编写的 S-Function

用 C 语言编写的 S-函数具有很多优点:执行速度快,可以实时生成代码, 可以包含已有的 C 代码,能够访问操作系统接口,还可以编写设备驱动程序. 所以本研究采用了 c 文件的形式编写 S-Function 代码. 它需要先编译成可以在 Matlab 内运行的二进制代码(动态链接库或者静态库),然后才能使用,这些经过编译的二进制文件即是所谓的 MEX 文件,在 Windows 系统下 MEX 文件后缀为 dll. 使用的命令为:mex-setup(选择编译器); mex my_sfunction. c(转换文件). 与编写 M 文件的 S-函数一样,Simulink 同样为用户提供了编写 C MEX S-函数所需的模板,文件 sfuntmpl_doc. c 则包含了所有的例程,并附有详细的注释.

2 建模过程

设机体坐标系相对平台坐标系的转动四元数为:

$$Q = q_0 + q_1 + ib + q_2 \cdot jb + q_3 \cdot kb$$

Q 的即时修正可通过解下面的四元数微分方程来实现:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -w^b_{p_{bx}} & -w^b_{p_{by}} & -w^b_{p_{bz}} \\ w^b_{p_{bx}} & 0 & w^b_{p_{bz}} & -w^b_{p_{by}} \\ w^b_{p_{by}} & -w^b_{p_{bz}} & 0 & w^b_{p_{bx}} \\ w^b_{p_{bz}} & w^b_{p_{by}} & -w^b_{p_{bx}} & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

* 收稿日期:2007-10-29

作者简介:赵芳(1981—),女,河北衡水人,硕士研究生,主要从事惯性导航系统的研究.

维普资讯 http://www.cqvip.com

由上式求出 q_0,q_1,q_2,q_3 可以得到捷联矩阵 T:

$$T=\begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2\cdot(q_1+q_2-q_0\cdot q_3) & 2\cdot(q_1\cdot q_3+q_0\cdot q_2)\\ 2\cdot(q_1\cdot q_2+q_0\cdot q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2\cdot(q_2\cdot q_3+q_0\cdot q_1)\\ 2\cdot(q_1\cdot q_2-q_0\cdot q_2) & 2\cdot(q_2\cdot q_3+q_0\cdot q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix}$$

加速度计测量的比力 f^b 通过矩阵 T 转换为 f^p,即

$$\begin{bmatrix} f_x^p\\ f_y^p\\ f_z^p \end{bmatrix}=\begin{bmatrix} T_{11} & T_{12} & T_{13}\\ T_{21} & T_{22} & T_{23}\\ T_{31} & T_{32} & T_{33} \end{bmatrix}\begin{bmatrix} f_x^b\\ f_y^b\\ f_z^b \end{bmatrix}$$

速度 $\dot{V}$ 的即时修正可通过解如下的微分方程来完成:

$$\begin{bmatrix} \dot{V}_x^p\\ \dot{V}_y^p\\ \dot{V}_z^p \end{bmatrix}=\begin{bmatrix} f_x^p\\ f_y^p\\ f_z^p \end{bmatrix}-\begin{bmatrix} 0\\ 0\\ g \end{bmatrix}$$

位置矩阵的即时修正:

$$\begin{bmatrix} \dot{C}_{11} & \dot{C}_{12} & \dot{C}_{13}\\ \dot{C}_{21} & \dot{C}_{22} & \dot{C}_{23}\\ \dot{C}_{31} & \dot{C}_{32} & \dot{C}_{33} \end{bmatrix}=\begin{bmatrix} 0 & 0 & -w_{epy}^p\\ 0 & 0 & -w_{epx}^p\\ w_{epy}^p & -w_{epx}^p & 0 \end{bmatrix}\begin{bmatrix} C_{11} & C_{12} & C_{13}\\ C_{21} & C_{22} & C_{23}\\ C_{31} & C_{32} & C_{33} \end{bmatrix}$$

姿态角的计算. 姿态角可以表示成 φ_G,θ,γ 的关系,即

$$T=\begin{bmatrix} \cos\gamma\cdot\cos\varphi_G-\sin\gamma\cdot\sin\theta\cdot\sin\cdot\theta\cdot\varphi_G & -\cos\gamma\cdot\sin\theta\cdot\varphi_G & \sin\gamma\cdot\cos\varphi_G+\cos\gamma\cdot\sin\theta\\ \cos\gamma\cdot\sin\varphi_G-\sin\gamma\cdot\sin\theta\cdot\cos\gamma\cdot\varphi_G & -\cos\theta\cdot\cos\gamma\cdot\varphi_G & \sin\gamma\cdot\cos\varphi_G+\cos\gamma\cdot\sin\theta\\ -\sin\gamma\cdot\cos\theta & \sin\theta & \cos\gamma\cdot\cos\theta \end{bmatrix}$$

由上式中所表示的矩阵 T 的元素 $T_{11},T_{22},T_{31},T_{32},T_{33}$ 可用下式计算 φ_G,θ,γ 的主值:

$$\theta_Z=\arcsin^{-1}T_{32}$$

$$\gamma_Z=\arctan^{-1}\frac{-T_{31}}{T_{33}}$$

$$\varphi_{GZ}=\arctan^{-1}\frac{-T_{12}}{T_{22}}$$

由 $\theta_Z,\gamma_Z,\varphi_{GZ}$ 判断其真值 θ,γ,φ_G 的公式是

$$\theta=\theta_Z$$

$$\gamma=\begin{cases}\gamma_Z & T_{33}>0\\ \begin{cases}\gamma_Z+180^\circ\\ \gamma_Z-180^\circ\end{cases} & T_{33}<0\quad\begin{cases}\gamma_Z<0\\ \gamma_Z>0\end{cases}\end{cases}$$

$$\varphi_G=\begin{cases}\varphi_{GZ} & \begin{cases}\varphi_{GZ}<0\\ \varphi_{GZ}>0\end{cases}\\ \begin{cases}\varphi_{GZ}+360^\circ\\ \varphi_{GZ}+360^\circ\end{cases} & T_{22}<0\quad T_{33}>0\end{cases}$$

位置计算. 位置矩阵可以表示成 φ,β,α 的关系,即

$$C=\begin{bmatrix} -\sin\alpha\cdot\sin\varphi\cdot\cos\beta-\cos\alpha\cdot\sin\beta & -\sin\alpha\cdot\sin\varphi\cdot\sin\beta+\cos\alpha\cdot\cos\beta & \sin\alpha\\ -\cos\alpha\cdot\sin\varphi\cdot\cos\beta-\sin\alpha\cdot\sin\beta & \cos\alpha\cdot\sin\varphi\cdot\sin\beta-\sin\alpha\cdot\cos\beta & \cos\alpha\\ \cos\varphi\cdot\cos\beta & \cos\varphi\cdot\sin\beta & \sin\alpha \end{bmatrix}$$

根据位置矩阵 C 的元素 $C_{13},C_{23},C_{31},C_{32},C_{33}$ 可由下式计算 φ,β,α 的主值:

$$\varphi_Z=\arcsin^{-1}C_{33}$$

$$\beta_Z=\arctan^{-1}\frac{C_{32}}{C_{31}}$$

$$\alpha_Z=\arctan^{-1}\frac{C_{13}}{C_{23}}$$

维普资讯 http://www.cqvip.com

由 $\varphi_Z,\beta_Z,\alpha_Z$ 判断其真值 φ,β,α 的公式是

$$\varphi = \varphi_Z$$

$$\beta = \begin{cases} \beta_Z & C_{31} > 0 \\ \begin{cases} \beta_Z + 180° \\ \beta_Z - 180° \end{cases} & C_{31} < 0 \quad \begin{cases} \beta_Z < 0 \\ \beta_Z > 0 \end{cases} \end{cases}$$

$$\alpha = \begin{cases} \alpha_Z & \begin{cases} \alpha_Z < 0 \\ \alpha_Z > 0 \end{cases} \\ \begin{cases} \alpha_Z + 360° \\ \alpha_Z + 360° \end{cases} & C_{23} < 0 \quad C_{23} > 0 \end{cases}$$

其中:下标 i 表示惯性系;e 表示地球系;t 表示地理系;b 表示机体系;p 表示平台系.

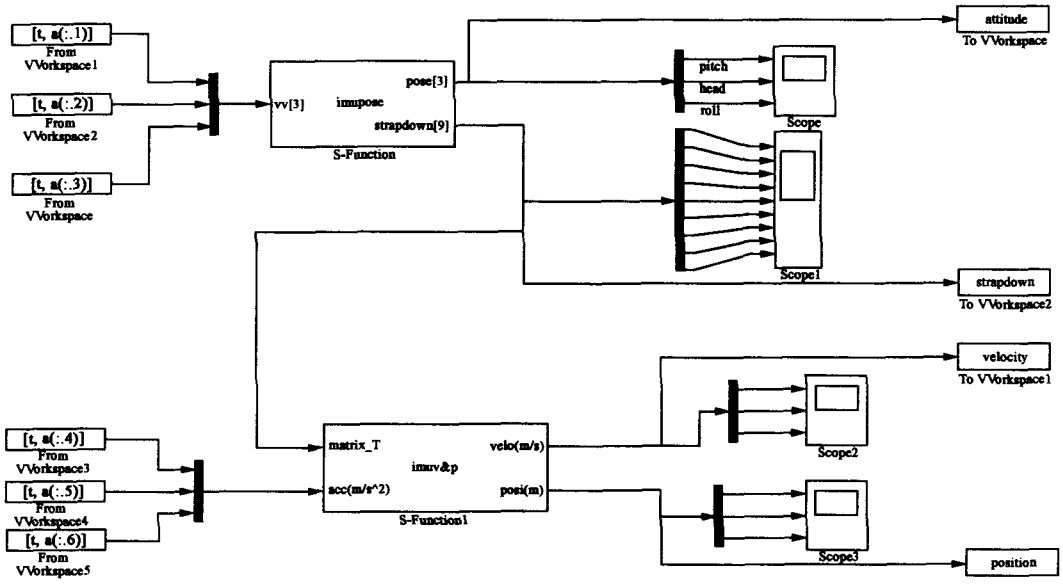


图 1 系统连接

系统连接如图 1 所示,系统建好之后就该编辑 S-Function 的程序代码了,新建一个 c 文件,拷入库中提供的 S-Function 的模板,在相应的地方修改和添加系统实际用到的输入、输出参数及相应的程序代码,其中第 1 个 S-Function 的初始化程序如下:

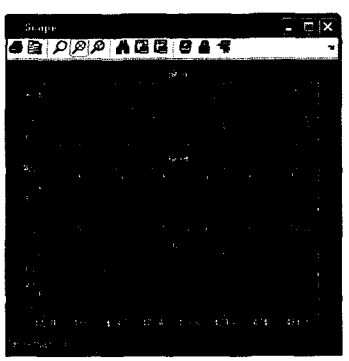
```
static void mdlInitializeSizes(SimStruct *S)
{ /* See sfuntmpl_doc.c for more details on the macros below */
    ssSetNumSFcnParams(S, 4); /* Number of expected parameters */
    if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount(S)) {
      /* Return if number of expected != number of actual parameters */
      return; }
  else
  {mdlCheckParameters(S); }
  {
       int iParam = 0;
       int nParam = ssGetNumSFcnParams(S);
       for ( iParam = 0; iParam < nParam; iParam++ )
       {ssSetSFcnParamTunable( S, iParam,
SS_PRM_SIM_ONLY_TUNABLE ); }
  }
  ssSetNumContStates(S, 0);
```

维普资讯 http://www.cqvip.com

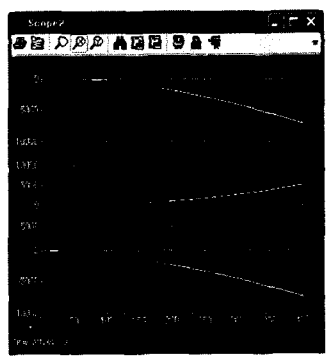

```
    ssSetNumDiscStates(S, 3);
    if (! ssSetNumInputPorts(S, 1)) return;
    ssSetInputPortWidth(S, 0, 3);
    ssSetInputPortRequiredContiguous(S, 0, true); /* direct input signal access */
    ssSetInputPortDirectFeedThrough(S, 0, 1);
    if (! ssSetNumOutputPorts(S, 2)) return;
    ssSetOutputPortWidth(S, 0, 3);
    ssSetOutputPortWidth(S, 1, 9);
    ssSetNumSampleTimes(S, 1);
    ssSetNumRWork(S,0);
    ssSetNumIWork(S, 0);
    ssSetNumPWork(S, 0);
    ssSetNumModes(S, 0);
    ssSetNumNonsampledZCs(S, 0);
    ssSetOptions(S, SS_OPTION_EXCEPTION_FREE_CODE);
}
```

3 仿真结果

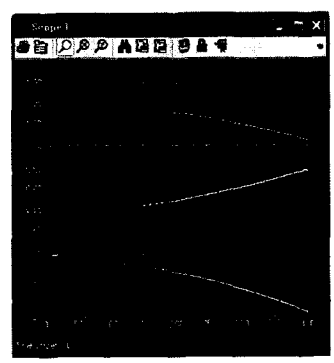
下面是某次试验的数据,仿真结果如图 2 所示. 然后再在 Matlab 命令窗口画出试验时导航计算机输出的速度、位置、姿态角的曲线,经比较,2 组曲线完全一致,证明模型仿真正确、可行.



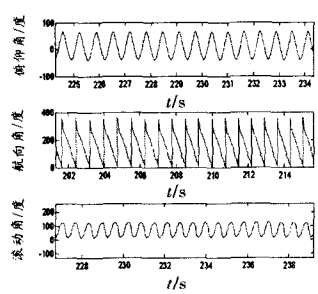
a) 仿真输出的三轴姿态角图线



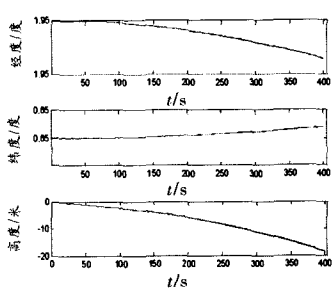
b) 仿真输出的三轴速度图线



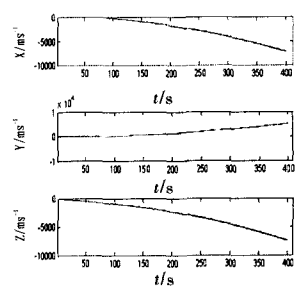
c) 仿真输出的三轴位置图线



d) 命令窗口输出的姿态角图线



e) 命令窗口输出的位置图线



f) 命令窗口输出的速度图线

图 2 仿真结果

参考文献:

[1] 陈杰. MATLAB 宝典[M]. 北京:电子工业出版社,2007:439 - 625.
[2] 陈桂明,张明照. 应用 MATLAB 建模与仿真[M]. 北京:科学出版社,2000:50 - 398.
[3] 崔中兴. 惯性导航系统[M]. 北京:国防工业出版社,1982:35 - 293.
[4] 陈哲. 捷联惯导系统原理[M]. 北京:宇航出版社,1986:9 - 212.