

# 基于北斗卫星导航系统的 UUV 遥控系统软件设计主控端

向小梅<sup>1</sup>, 聂林波<sup>2</sup>

(1. 91388 部队, 广东 湛江 524022; 2. 南海舰队海洋水文气象中心, 广东 湛江 524001)

**摘要:**针对目前 UUV 通信控制现状,提出了基于北斗导航系统的 UUV 遥控系统设计构想;采用北斗导航系统的导航定位和短报文通信功能实现 UUV 遥控系统软件设计;系统可迅速确定 UUV 位置信息,远程传递各种控制指令和 UUV 探测的信息以及 UUV 自身状态数据,增强了地面指挥中心对 UUV 的遥控能力并拓展了 UUV 的活动范围;基于“北斗”导航系统的 UUV 遥控系统为 UUV 管理控制功能国产化提供了一个新的发展方向。

**关键词:**北斗卫星导航系统;水下无人航行器;遥控;指挥控制

**本文引用格式:**向小梅,聂林波.基于北斗卫星导航系统的 UUV 遥控系统软件设计主控端[J].四川兵工学报,2015(5):42-45.

**Citation format:**XIANG Xiao-mei, NIE Lin-bo. Design for Remote Control System of UUV Based on Beidou Satellite Navigation System[J]. Journal of Sichuan Ordnance, 2015(5):42-45.

中图分类号:TJ67

文献标识码:A

文章编号:1006-0707(2015)05-0042-04

## Design for Remote Control System of UUV Based on Beidou Satellite Navigation System

XIANG Xiao-mei<sup>1</sup>, NIE Lin-bo<sup>2</sup>

(1. The No. 91388<sup>th</sup> Troop of PLA, Zhanjiang 524022, China; 2. The Ocean Meteorological and Hydrological Center of South China Sea Fleet, Zhanjiang 524001, China)

**Abstract:** To improve the remote control system of UUV, a remote control system of UUV based on the Beidou satellite navigation system was proposed. The navigation and short message communication function of the Beidou satellite navigation system was adopted to realize the remote control system. The remote control system can not only position UUV rapidly, but also transfer commands and data, which enhance the remote control ability of ground control center and expand the scope of activities of UUV. The remote control system of UUV Based on the Beidou satellite navigation system provides a new direction of development of remote control of UUV.

**Key words:** Beidou satellite navigation system; UUV; remote control; command and control

UUV(水下无人航行器)主要通信手段为水声通信。UUV多数时间在水面以下,无法进行无线电通信,一旦UUV浮出水面,就可以利用卫星进行无线电通信。我国自主研发的北斗卫星导航系统可以提供定位和短报文通信功能。UUV的遥控指挥可以借助北斗导航系统实现。这样的优点是能在更远海域进行UUV的定位和控制,同时降低了系统

设计和维护成本,系统更加稳定可靠。

### 1 北斗卫星系统

北斗卫星导航系统是我国独立研制的集测量技术、定位技术、数字通信和扩频技术为一体的全天候区域性有源三维

卫星导航、定位、通信系统<sup>[1-4]</sup>。它是除美国的GPS、俄罗斯的GLONASS之后第三个成熟的卫星导航系统。该系统于2000年初步建成,2003年实现了全网运营,主要用于国家经济建设,为我国的交通运输、气象、石油、海洋、森林防火、灾害预报、通信、公安以及其他特殊行业提供高效的导航定位及通信服务。北斗卫星系统由四颗北斗卫星(两颗备用)、地面控制中心和北斗用户终端三部分组成,具有快速定位、双向短报文通信和精密授时服务,其覆盖范围是北纬 $5^{\circ} \sim 55^{\circ}$ ,东经 $70^{\circ} \sim 140^{\circ}$ ,上大下小,最宽处在北纬 $35^{\circ}$ 左右,覆盖我国的全部国土,以及印度洋和太平洋相当广阔的一部分。在定位方面,该系统能快速定位用户地理位置,为用户及主管部门提供导航;水平定位精度100 m,差分定位精度小于20 m;定位响应时间:1类用户5 s,2类用户2 s,3类用户3 s,简短定位更新时间小于1 s,一次性定位成功率95%。在通信方面,北斗系统专门开发出军用通信频段,以支持广大军队用户使用,最多一次可传输120个汉字,可以实现用户与用户、用户与地面控制中心之间通信<sup>[5-8]</sup>。

## 2 基于北斗导航系统的UUV遥控系统工作原理

UUV处于水面工作状态时接收北斗导航卫星传输的数据,UUV的遥控指令处理模块按照遥控协议对数据进行处理,提取指令信息,将指令发给UUV航行控制模块,UUV航行控制模块按照指令执行动作,实现UUV的运动控制;UUV航行控制模块将自身状态信息反馈给状态报告模块,状态信息按照遥控协议编码后经北斗通信模块发送给主控端。主控端接收UUV经北斗导航卫星传输的数据,按照遥控协议处理数据,提取UUV状态信息,UUV控制决策模块根据UUV的信息制定下一步控制指令。整个系统的控制流程如图1所示。

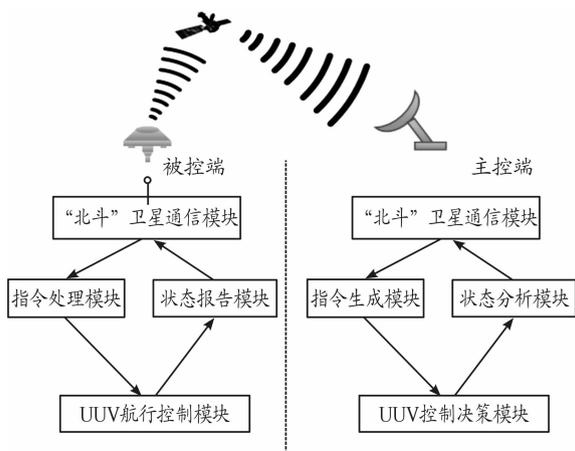


图1 基于北斗导航系统的UUV遥控系统控制流程

## 3 遥控系统软件设计

系统软件由主控端和被控端两部分组成。主控端发送

控制指令给被控端,接收被控端传回的应答信息,根据返回的信息发送后续指令;被控端接收主控端的控制指令,根据指令执行相应的动作,向主控端发送自身状态信息,向主控端报告指令执行情况。主控端能实时显示UUV的状态信息、位置信息、指令执行情况。主控端利用电子海图实现图形化显示和指挥。当UUV在水面工作时,北斗通信模块启动,将UUV传感器采集环境数据和UUV自身工作状态按照遥控协议预先设定的格式经北斗导航卫星传输到主控端。主控端将UUV的当前位置、状态和轨迹等信息以图形化的方式现在电子海图上。主控端根据UUV的状态信息自动或由人工判别并下发后续控制指令,控制UUV的动作、姿态、运动方向和移动距离等。主控端显示软件基于电子海图平台。电子海图用来实时显示UUV状态信息,记录UUV的航行轨迹。主控端可以控制多批UUV,向每个UUV发出不同指令,控制UUV的运动。主控端也可以对UUV进行编组,实现同一组UUV一致动作。

### 3.1 遥控协议设计

UUV遥控系统可由一个主控端和多个被控端通过北斗导航卫星系统互连而成。要使整个遥控系统有条不紊地工作,就要求每个终端必须遵守一些事先约定好的有关数据格式及时序等的规则。这些为实现遥控而建立的规则、约定就是遥控协议。遥控协议由3个要素<sup>[9]</sup>组成:语法、语义和时序。语法即控制指令和状态信息的结构或格式;语义即需要发出何种控制指令,完成何种动作以及做出何种响应;时序即遥控过程实现顺序的详细说明。

一套功能完善的遥控协议能确保系统可靠地实现对UUV的控制功能。遥控协议需要解决的问题:一是对指令集进行规范说明,确保指令的完备性满足UUV控制需要。指令集大致分为动作指令、状态反馈指令、定位指令、时间统一指令。二是需要对指令和信息的格式进行规范说明,确保双方能正确理解指令和信息。三是需要包含时间统一机制和指令延迟处理机制,需要解决信息传输时间延迟对UUV控制的精度影响和信息受干扰丢失对UUV控制的影响等问题。

### 3.2 软件设计关键技术

根据遥控系统的功能需求,遥控软件主要用到GIS技术、串口通信技术和多线程技术。

系统基于GIS平台,能实现地图显示、计算和分析等功能。本系统采用TopMap ActiveX作为GIS平台。TopMap ActiveX是组件式地理信息系统开发平台,产品包括TopMap ActiveX主控件、全图浏览定位的鸟瞰图控件、属性列表编辑控件、地图图层列表及管理控件等,适用于单机及局域网、客户端/服务器模式的地理信息系统应用开发。在面向对象的可视化编程语言(如VB、VB.NET、C#、VC等)中插入TopMap ActiveX组件,就可实现地理信息系统功能<sup>[10]</sup>。

安装完TopMap ActiveX后,即可在Visual Studio2010创建的项目中引入TopMap ActiveX组件。在Visual Studio2010的开发界面里选择工具箱/组件,在空白处右键,选择“选择项...”。选择COM组件下的TopMap ActiveX GIS Control

Module V6。在工具箱的组件下就会出现 TopMap ActiveX GIS Control Module V6。鼠标选中 TopMap ActiveX GIS Control Module V6,在窗体中按下左键拖动鼠标绘制大小合适的矩形,松开左键,就在窗体中添加了 TopMap 地图组件。

加载地图用下面的代码实现:

```
TGMAPX1. MapLoad( "d:\maps\Map06. TGF" );
```

显示地图:

```
TGMAPX1. DrawMapFull();
```

设置某个图层为动态图层后,由于动态图层提供了绘制图层(DrawMapDynamic)的方法,而不需要重新绘制整个地图,所以加快了地动态图层刷新图的绘制速度。对某个图层进行了实体添加操作时,在添加实体的同时会在地图上看到添加的实体,而这时显示范围是不会发生变化的,在这种情况下,就可以采用刷新动态图层的绘制方法。除添加实体外,实体移动、实体闪烁等等都可以采用动态图层方法。地图视野范围发生改变,如放大、缩小等整个地图视野发生改变,在这种情况下不可以采用动态图层刷新。

```
TGMAPX1. LayerSet( 0, TGMAPXLib. emLayerFlag. ET-GLayerIsDynamic, True );
```

```
TGMAPX1. DrawMapDynamic();
```

用 TGOFeatureSymbol 绘制 UAV 目标(点实体对象)。

```
TGOFeatureSymbol pFeatureSymbol = new TGOFeatureSym-  
bolClass();
```

```
TGORecordSet RecordSet = new TGORecordSetClass();
```

//设置点实体样式

```
pFeatureSymbol. SymStyle. ForeColor. R = 125;
```

```
pFeatureSymbol. SymStyle. ForeColor. G = 240;
```

```
pFeatureSymbol. SymStyle. ForeColor. B = 255;
```

```
pFeatureSymbol. SymStyle. Lib = 0;
```

```
pFeatureSymbol. SymStyle. index = 4;
```

```
pFeatureSymbol. Symbol. x = lon;
```

```
pFeatureSymbol. Symbol. y = lat;
```

```
RecordSet. AddNew();
```

```
RecordSet. feature = pFeatureSymbol;
```

```
RecordSet. Close();
```

```
TMAPX1. DrawMapFull();
```

用 TGOFeaturePolyline 绘制 UAV 航线轨迹。

```
TGOFeaturePolyline pFeaturePolyline = new TGOFeature-  
PolylineClass();
```

```
TGOPoints pts = new TGOPointsClass();
```

```
TGOPoint pt = new TGOPointClass();
```

```
pt. x = lon1;
```

```
pt. y = lat1;
```

```
pts. Add( -1, pt );
```

```
pt. x = lon2;
```

```
pt. y = lat2;
```

```
pts. Add( -1, pt );
```

//把点序列添加到线

```
pFeaturePolyline. polyline. parts. Add( -1, pts );
```

```
RecordSet. AddNew();
```

```
RecordSet. feature = pFeaturePolyline;
```

```
RecordSet. Close();
```

```
TMAPX1. DrawMapFull();
```

在该系统中采用 TopMap ActiveX 组件实现各种 GIS 功能包括 UAV 定位、航迹显示、距离计算、速度推算、位置推算等功能。

计算机通过串口与北斗终端连接。指令和数据均通过串口来传输。Visual Studio2010 中的命名空间 System. IO. Prots 提供了 SerialPort 类来实现对串口的访问。SerialPort 类支持同步 I/O 和事件驱动 I/O。

使用该类需要先引用:using System. IO. Ports;

创建串口对象:mserialPort = new SerialPort();

设置串口属性值:

```
mserialPort. PortName = "com1";
```

```
mserialPort. BaudRate = 9 600;
```

```
mserialPort. Parity = Parity. None;
```

```
mserialPort. DataBits = 8;
```

```
mserialPort. StopBits = StopBits. One;
```

```
mserialPort. Handshake = Handshake. None;
```

设置串口读写超时值:

```
mserialPort. ReadTimeout = 500;
```

```
mserialPort. WriteTimeout = 500;
```

打开串口;mserialPort. Open();

获取当前计算机的串口名称数组:

```
mserialPort. GetPortNames();
```

读写串口;mserialPort. read(),mserialPort. write()

系统要具有多任务、多线程和实时监控功能。C#能很好实现多任务和多线程编程。C#能实现操作系统级别的实现数据监测功能,提供了较高的实时性。

在 C#中 system. threading 命名空间提供了进行多线程编程的类和接口。常用的实现多线程方法有 thread、threadpool 和 timer。另外 C#中的 backgroundworker 组件提供了最简单的一种多线程解决方法<sup>[11]</sup>。

Thread 的用法如下所示:

```
using System. Threading;
```

```
Thread newThread = new Thread(new ThreadStart(work));
```

```
newThread. Priority = ThreadPriority. BelowNormal;
```

```
newThread. Start();
```

```
public static void work()
```

```
{
```

```
//耗时的计算
```

```
}
```

priority 指定一个线程相对于另一个线程的相对优先级。

ThreadState 定义线程组所有可能的执行状态。一旦线程被创建,它就至少处于其中一个状态中,直到终止。

threadpool 方法相对 thread 较简单,它适应于一些需要多个线程而又较短任务,它的缺点是对创建的线程不能加以控制,也不能设置其优先级。threadpool. queueuserworkitem(new

WaitCallback(ThreadProc))将方法排入队列等待线程池线程变得可用时执行。

timer 以指定的时间间隔执行方法。使用 TimerCallback 委托指定 Timer 执行的方法。计时器委托在构造计时器时指定,并且不能更改。此方法不在创建计时器的线程上执行,而是在系统提供的 threadpool 线程上执行。构造计时器时可以指定在第一次执行方法之前等待的时间以及此后的执行期间等待的时间。Change 方法用于更改这些值或禁用计时器。

C#中的 backgroundworker 组件提供了最简单的一种多线程解决方法<sup>[11]</sup>。该组件在应用程序的界面线程的后台执行耗时的操作。上传和下载、访问数据库、读写磁盘文件等需要很长执行时间的操作可能导致用户界面在操作运行时无法响应。BackgroundWorker 组件可以解决这些问题。使用 BackgroundWorker 组件,先定义要在后台执行的辅助方法,然后调用 RunWorkerAsync 方法。在辅助方法以异步方式运行的同时,界面线程继续正常运行。该方法运行完毕,BackgroundWorker 激发 RunWorkerCompleted 事件向调用线程发出警报。

BackgroundWorker 类的 DoWork 事件处理程序采用一个 DoWorkEventArgs 参数,该参数具有 Argument 属性。此属性接收来自 RunWorkerAsync 的参数,并可以传递至 DoWork 事件处理程序中调用的辅助方法。

C#中的 FileSystemWatcher 组件提供对文件的实时监控<sup>[11]</sup>。使用 FileSystemWatcher 监视指定目录中的更改。可监视指定目录中的文件或子目录的更改。可以创建一个组件来监视本地计算机、网络驱动器或远程计算机上的文件。Filter 属性设置为空字符串("")或使用通配符("\*.\*")可以监视所有文件中的更改,将 Filter 属性设置为某个文件名即可监视该文件。将 Filter 属性设置为 "\*.txt"即可监视特定类型文件中的更改。监视文件或目录的 Attributes、LastWrite 日期和时间或 Size 方面的更改可通过将 NotifyFilter 属性设置为 NotifyFilters 枚举中的某个或某一组值来实现。Windows 操作系统在 FileSystemWatcher 创建的缓冲区中通知组件文件发生更改。如果短时间内有很多更改,则缓冲区可能会溢出。这将导致组件失去对目录更改的跟踪。为防止缓冲区溢出,可使用 FileSystemWatcher.NotifyFilter 和 FileSystemWatcher.IncludeSubdirectories 属性筛选掉不想要的更改通知。也可以通过 FileSystemWatcher.InternalBufferSize 属性增加内部缓冲区的大小。但缓冲区来自无法换出到磁盘的非页面内存,增加缓冲区大小的成本较大,所以将缓冲区保持大小适中,以便不会丢失任何文件更改事件。

在该系统的设计中采用 backgroundworker 组件实现对 UUV 的指令发送、位置推算等操作,采用 FileSystemWatcher 组件实现 UUV 回传数据的监测和处理操作,采用 thread 类实现多批 UUV 目标的定位和通信。

## 4 结束语

基于“北斗”导航系统的 UUV 遥控系统为 UUV 管理控制功能国产化提供了一个新的发展方向。由于 UUV 遥控系统依托北斗导航系统实现控制指令和感知信息的传递和协同,既可迅速确定 UUV 位置信息,又可远程传递各种控制指令和 UUV 探测的信息以及 UUV 自身状态数据,极大地增强了地面指挥中心对 UUV 的遥控能力并拓展了 UUV 的活动范围。

## 参考文献:

- [1] 徐常凯,郭瑞勇,郭军.基于北斗卫星的军事物流通信系统研究[J].物流技术,2009,28(9):147-149.
- [2] 江彩英,黄永玉,谢丹,等.北斗卫星应急通信传输试点建设[J].气象科技,2013,41(5):857-860.
- [3] 庞佑军.基于北斗卫星导航的搜救系统原理与构型[J].航空电子技术,2013(4):7-8.
- [4] 唐金元,于璐,王思臣.北斗卫星导航定位系统应用现状分析[J].全球定位系统,2008(2):26-27.
- [5] 于天泽.北斗卫星导航定位技术在我国铁路应用探讨[J].中国铁路,2013(4):4-5.
- [6] 吴才聪,苏怀洪,褚天行,等.基于北斗的移动应急监控与指挥技术[J].数字通信世界,2011(12):60-61.
- [7] 李经.基于 AIS 的舰艇编队避碰技术研究[J].舰船电子工程,2010(8):69-70.
- [8] 于光峰.舰载 AIS 信息采集与解码技术研究[J].电子技术与软件工程,2013(21):91-92.
- [9] Andrew S. Tanenbaum. Computer networks[M]. 北京:清华大学出版社,2001:17-20.
- [10] 北京慧图信息科技有限公司. TopMap ActiveX 开发帮助[EB/OL]. <http://www.topmap.com.cn>,2011.
- [11] Microsoft. Microsoft Help Library[EB/OL], <http://www.microsoft.com/zh-cn/download/developer-tools.aspx>,2010.

(责任编辑 杨继森)