M.Sc. in SPACE STUDIES 2005/2006

INTERNATIONAL SPACE UNIVERSITY

Upgrade of Satellite Receiver Localisation System

Seppo Heikkilä

Individual Project Report submitted to the International Space University in partial fulfillment of the requirements of the M.Sc. Degree in Space Studies

August, 2006

Internship Mentor:	Guy Harles and Markus Gross
Host Institution:	SES Astra SA, Betzdorf, Luxembourg
ISU Academic Advisor:	Dr. Isabelle Scholl

ACKNOWLEDGEMENTS

This individual project has been carried out at the Ground Station and Solutions section at SES Astra SA during the summer 2006 as fulfillment for Master of Science in Space Studies degree at International Space University (ISU).

I want to thank the European Space Agency, Automation Technology Laboratory of Helsinki University of Technology, and SES Global for the generous support towards my studies at the ISU and for the possibility to do my internship at the SES Astra SA in Betzdorf, Luxembourg.

Special thanks to Markus Gross and Guy Harles, my instructors at the SES Astra, for their continuous support for this report. I want to thank also my academic advisor at ISU, Dr. Isabelle Scholl, for the feedback given for the report.

Seppo Heikkilä
Betzdorf, Luxembourg August 11th 2006

ABSTRACT

This report presents an upgrade to a satellite receiver localization box, named ZipMode box. It is designed to localize satellite receivers with few kilometer accuracy using Digital Video Broadcasting over Satellite (DVB-S) satellite signals. The Zipmode box is developed to be affordable for mass markets so it has to cope with scarce hardware resources and small return channel bandwidth.

Order to develop a stand-alone ZipMode box, an embedded Linux operating system is set up on a PC/104 386SX computer equipped with 2MB of RAM. The usage of open source Linux provides a large set of software development tools, device drivers and miscellaneous applications. Order to make the development fast and convenient, the operating system and ZipMode application code are cross-compiled using a standard Linux desktop computer. In this study the embedded Linux distribution demonstrated to be suitable for the ZipMode box is Small Linux, which is based on the old Linux 2.0 kernel.

To enhance the ZipMode functionality, a literature review of currently existing satellite localization methods is conducted. The focus is on the orbit determination methods and on the satellite orbit modelling. The literature review shows that the current orbit determination techniques can achieve subcentimeter level orbital precisions when combined with satellite orbit models. The Precise Orbit Determination (POD) is a direct prerequisite for accurate satellite receiver losalization.

TABLE OF CONTENTS

INDEX OF FIGURES

INDEX OF TABLES

LIST OF ACRONYMS

AC/DC	Alternating Current/Direct Current
BILSAT	BILSAT, Turkey's first remote sensing satellite
BBF	Bayesian Bootstrap Filter, recursive SIR
CD	Compact Disc
CF	Compact Flash
CHAMP	CHAllenging Minisatellite Payload, German research satellite
CNES	Centre National d'Etudes Spatiales, French space agency
DARTS	Digital Advanced Ranging with Transport-stream, ranging system by SES Astra
DGPS	Differential GPS
DIODE	Doris Immediate On-board orbit DEtermination, RT orbit determination software
DORIS	Doppler Orbiting and Radiopositioning Integrated by Satellite
DRR	Dual Ranging Receiver
DVB-S	Digital Video Broadcasting over Satellite
Envisat	Environmental Satellite, ESA's Earth-observing satellite
ESA	European Space Agency
GCSP	Geostationary Colocated Satellite Positioning
GDOP	Geometrical Dilution of Precision, dimensionless position accuracy ratio
GEO	Geostationary Earth Orbit
GIPSY	GPS Inferred Positioning SYstem, JPL orbit determination software
GNSS	Global Navigation Satellite System
GPL	General Public License
GPS	Global Positioning System, United States NAVSTAR GPS
GRUB	GRand Unified Bootloader, versatile bootstrap loader
IGS	International GPS Service
IMU	Inertial Measurement Unit
iTV	Interactive Television
JPL	Jet Propulsion Laboratory
LBS	Location Based Services
LEO	Low Earth Orbit
LGPL	Lesser General Public License
LoS	Line of Sight

LRR	Laser Retroreflector, directly back reflecting prism
LSE	Least Square Estimation, mathematical optimization technique
MANS	Microcosm Autonomous Navigation System
MB	Megabyte
MBR	Master Boot Record
NASA	National Aeronautics and Space Administration, United States space agency
NAVSAT	Navy Navigation Satellite System, a.k.a. TRANSIT
NAVSTAR	NAVigation Signal Timing And Ranging
POD	Precise Orbit Determination
PRARE	Precise Range And Range-Rate Equipment
RAM	Random Access Memory
RMS	Root Mean Square, measure of magnitude of varying of values
RT	Real-Time
RTD	Round Trip Delay
S/A	Selective Availability
SIR	Sampling Importance Resampling
SLR	Satellite Laser Ranging
SPOT	Système Pour l'Observation de la Terre, French remote sensing satellite
STEP	Space Test Experiment Platform
T/P	TOPEX/POSEIDON
TAOS	Technology for Autonomous Operational Survivability, USAF's technology demonstration satellite
TDOA	Time Difference Of Arrival
TDRSS	Tracking and Data Relay Satellite System, NASA's signal relay system
TOF	Time of Flight
TV	Television
USAF	United States Air Force

1 INTRODUCTION

Global Positioning System (GPS) receivers are considered today standard products for reliable localization. They provide fast and relatively accurate localization around the globe for various applications. For example GPS based car navigators are now emerging with a speed that could make them as common appliances as cell phones within a decade. The GPS localization information can be further associated with other information to provide Location Based Services (LBS), such as shopping and leisure notices.

One significant problem of GPS is that it requires a line of sight to the satellites. This means that GPS is not usable indoors. For this reason the GPS receivers have not been integrated with home electronics, such as home theaters. For example satellite receivers could utilize the localization information for determining automatically the country settings and available regional programs. Furthermore, this localization information could be also transmitted to central server where it can be used to identify important customer areas and the associated connection qualities.

1.1 Satellite Receiver Localization

Solution for the lack of cost-effective GPS localization can be found for satellite receivers from the fact that they have antennas fixed outside with good line of sight to the satellites, see Figure 1-1. The satellite receivers have already the needed hardware to receive and process the satellite signals. Satellite receiver does not receive signal from the localization satellites but from the Television (TV) broadcasting Geostationary Earth Orbit (GEO) satellites. The localization principles of localization satellites, such as GPS, can still be utilized for these satellites.

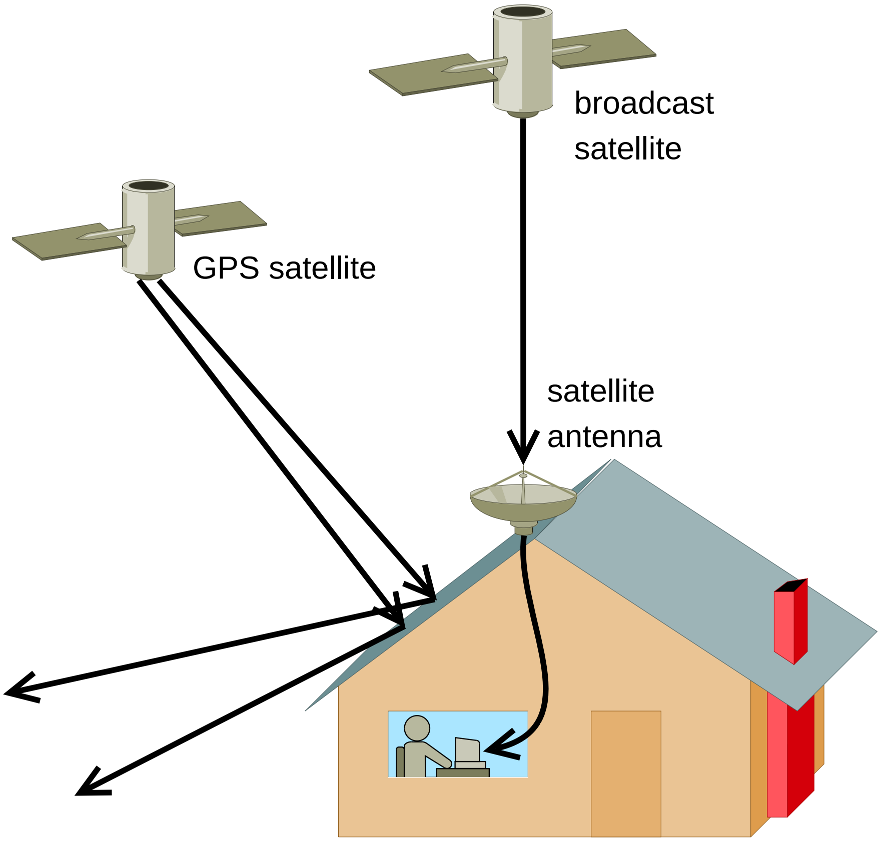

Figure 1-1: Broadcast satellites enable indoor satellite receiver localization

One of the major challenges of using TV broadcast satellites for localization is that the usable satellites are collocated at the GEO. The huge distance to GEO and small distance between the satellites creates a challenging combination. It has been actually shown that accurate instantaneous positioning is impossible with these given constraints, but instead long time observations, using the small motion of satellites, can provide an usable solution [Gross et al., 2006a].

This report presents an upgrade to the Geostationary Colocated Satellite Positioning (GCSP) receiver, called ZipMode, presented in [Gross et al., 2006a]. The ZipMode GCSP system uses two collocated satellites, one uplink ground station and a ZipMode satellite receiver, as shown in Figure 1-2. The satellite receiver box measures the Time Difference Of Arrival (TDOA) of special ranging packets that are inserted to the transport stream. The advantage of measuring TDOA is that the uplink ground station and satellite receiver clocks do not have to be synchronized.

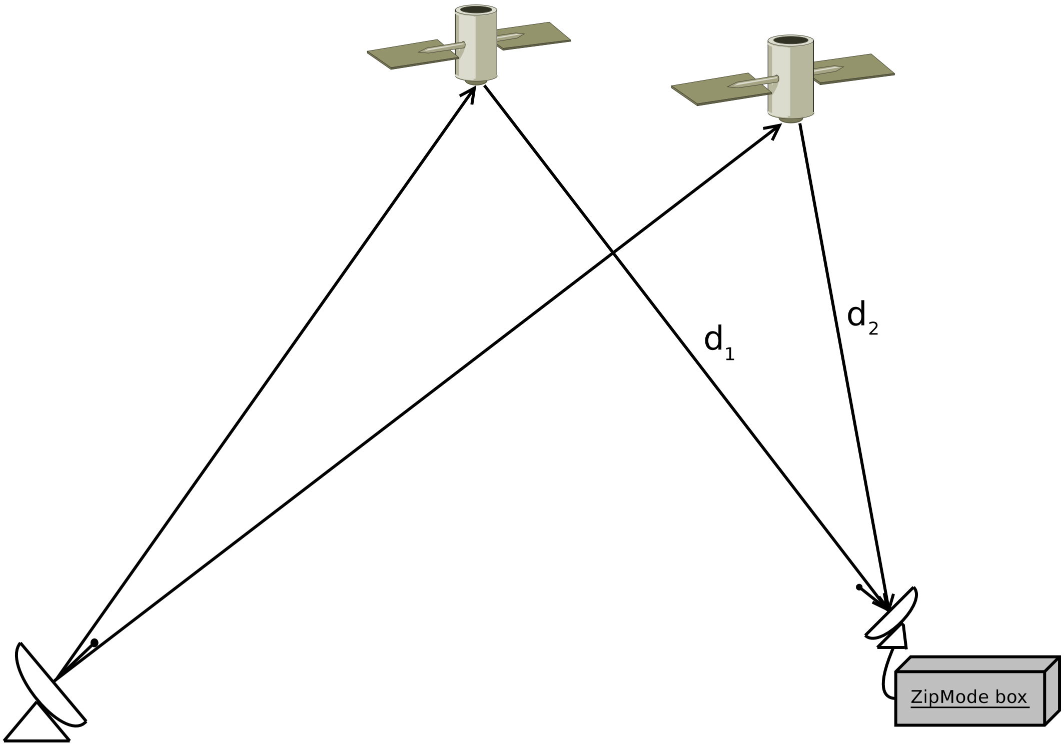

Figure 1-2: ZipMode system operation principle

The TDOA measurements are used to estimate the range difference between satellites, i.e. the $\Delta d = d_1 - d_2$ shown in Figure 1-2. The satellite locations are assumed to be known as well as the location of the ground station. This means that the only unknown factors are the distances from the satellites to the receiver, i.e. d_1 and d_2. These distances cannot be measured with this setup, but the range difference Δd can be determined using the TDOA measurements

The proposed ZipMode system, described with details in [Gross et al., 2006a,b,c], is developed under European Space Agency (ESA)'s SATMODE project. The SATMODE project aims to find a low cost Interactive Television (iTV) solution for mass markets via satellites. The system is demonstrated to be able to localize within a 1.5km radius circle for 50% of the time, and to localize the satellite receivers within the whole satellite footprint. These results were achieved by taking three range difference measurements per minute over period of 12 hours in each measurement location.

2 ZIPMODE SATELLITE RECEIVER LOCALIZATION BOX

The object of the first part of this study is to upgrade a developed satellite receiver box from DOS based operating system to Linux based operating system. By doing this it should be possible to

- Find out easily application memory requirements
- Run parallel processes, e.g. simultaneous serial port reading and data processing
- Run parallel process to slow down the processor speed

The upgrade is realized by selecting a suitable Linux distribution, converting and compiling the already existing DOS based application to run on it, and then verifying the system using real satellite data. The system booting and distribution selection are described in Section 2.1 and 2.2, the compilation process in Section 2.3 and 2.4, and the test results at the end in Section 2.5.

The embedded satellite receiver localization box referred in this study is called ZipMode and it is introduced earlier in [Gross et al., 2006a,b,c]. The [Gross et al., 2006a] describes the operation principles and test results of the receiver, [Gross et al., 2006b] focuses on the Zipmode hardware and algorithms, and [Gross et al., 2006c] presents the ZipMode box capabilities after a system upgrade. A prototype of ZipMode box is shown in Figure 2-1.

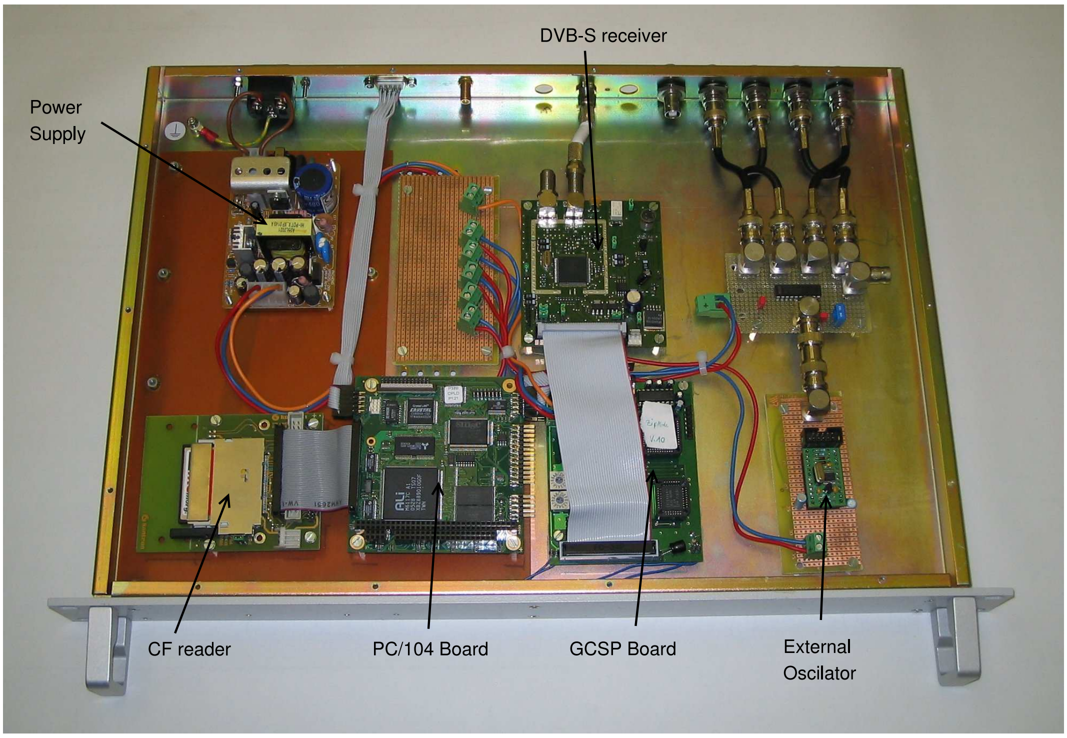

Figure 2-1: Hardware of prototype ZipMode box

The ZipMode box prototype hardware consist of

- DVB-S signal receiver
- GCSP board (processes the received data stream)
- Power supply (AC/DC converter)
- PC/104 board (Kontron MOPS/386A) with
  - 386SX 40MHz microprocessor

  - 2MB of on-board memory
  - two RS232 ports
  - one parallel port
  - 10Mbit CS8900A Ethernet controller

- Compact Flash (CF) card reader (with 256MB CF disk)
- External oscillator board (optional)

The ZipMode PC/104 board sets the constraints for the operating system selection. The most significant constraint is the amount of available on-board memory. Most of the Linux distributions are developed for modern computers with tens of megabytes of memory. For this reason the suitable distributions are old ones or specifically targeted for embedded systems.

The requirements defined for the target ZipMode box Linux distribution are

- Multitasking support (several concurrent processes)
- Embedded operating system, no Real-Time (RT) requirements
- Serial interface support
- File system support
- Floating point emulation
- TCP/IP support (optional)
- Command shell (optional)

The above described ZipMode box computer is referred in this document as a "target". The host computer, i.e. the development computer, is assumed to have a Linux distribution with standard GNU C compiler and tools. In this study the host computer is 64-bit Debian 4.0.3 running a Linux 2.6.17 kernel. The cross-compilation is done on a chroot environment with the current stable (sid) 32-bit Debian. For all the examples shown the CF device is assumed to reside in /dev/sda.

2.1 Making CF Bootable System

The first step when testing different operating systems is to make the target bootable. This can be achieved with boot loader which is small program that can be used to load different operating systems from disk to the memory. Loading an operating system is complex task, requiring drivers and software tools, and thus in the development phase a versatile boot loader is almost compulsory requirement.

The boot loader, e.g. GRand Unified Bootloader (GRUB), has to be installed on the first sector of your bootable media. For this reason it is advised to backup the first sector of the disk on which you are installing your boot loader. The backup and restore of the Master Boot Record (MBR) can be done as shown in Listing 2.1. If you want to restore only the MBR, and e.g. create the partition table again manually, you can write back only the first 446 bytes.

Listing 2.1: Backup and restore of MBR

```
# Backup the MBR and partition table
dd if=/dev/sda bs=512 count=1 of=MBR.dd
# Restore the MBR and partition table
dd if=MBR.dd of=/dev/sda bs=512 count=1
```

```
# Restore only the MBR
dd if=MBR.dd of=/dev/sda bs=446 count=1
```

2.1.1 Booting without Filesystem

The most straightforward way to make a CF bootable is to write the boot loader directly to the disk. This is easy because only the boot loader binary suitable for the target architecture is needed. Any computer can be used to write the binary files to the disk. For example the GRUB boot loader can be installed in Linux as shown in Listing 2.2.

Listing 2.2: Installing the boot loader

```
# Write GRUB boot loader to the disk
cat stage1 stage2 | dd if=/dev/sda
```

The major drawback of this approach is that the CF does not have now file system and thus another media is needed to provide the root file system and the kernel image.

2.1.2 Booting with File system

In practise all usable systems have a file system where the operating system and executable files are stored. Without a file system the development process would be much slower. In this section the GRUB is set up on the CF with a file system support.

Creating a File system

First step is to partition the CF for wanted number of file systems. Usually two is enough, one root and one swap partition. Only the root partition is compulsory, swap is only needed to provide additional memory. Partitioning can be done with command shown in Listing 2.3, which has additional -z parameter to ignore the current partition table if one already exists.

Listing 2.3: Creating partition table on the CF

```
# Partition /dev/sda, ignore existing partition table
cfdisk -z /dev/sda
```

With the cfdisk, select the partition type Linux Swap (82) and Linux (83), and mark the bootable flag on for Linux (83) partition. Finally write the partition table to the disk.

Next step is to initialize the file system of the Linux partition, e.g. partition /dev/sda2. This can be done as shown in Listing 2.4. The “-O none” parameter assures that the file system is compatible with old kernels, i.e. no extra features will be included.

Listing 2.4: CF with an ext2 file system and swap

```
# Initialize swap partition
mkswap /dev/sda1
# Format the second partition as ext2 file system
mkfs.ext2 -O none /dev/sda2
```

Installing the Boot Loader

Next a boot loader has to be installed to the CF MBR. Currently most versatile and compatible boot loader is currently GRUB. It is also default boot loader of Debian Linux distribution.

One important requirement is to run the GRUB from a computer with a same architecture than the target computer. For example GRUB run through 64-bit Debian did not work in the target i386 machine. For this reason a bootable Live CD, i.e. a operating system usable from CD without installation, might be necessary for the boot loader installation. For example Knoppix Linux Live CD can be downloaded for this purpose. In this study the Knoppix version 4.0 was used to set up the GRUB.

Before running the grub you need to copy boot loader files to the file system you created in Section 2.1.2. The required files are shown in Table 2-1.

Table 2-1: GRUB boot loader files

File	Description
stage1	MBR boot image (size 512 bytes) used to boot up GRUB
stage2	The GRUB core file that is loaded by stage1 or stage1_5
stage1_5	Same as stage1 but understands file systems (optional)
menu.lst	GRUB configuration file (optional)

These GRUB files can be copied from the installation platform, e.g. from the Knoppix Live CD, or downloaded from ftp://alpha.gnu.org/gnu/grub/. The files have to be copied at their place in the CF as shown in Listing 2.5.

Listing 2.5: Copying the GRUB files to the CF

```
# Mount CF to /mnt/cf
mkdir /mnt/cf
mount /dev/sda2 /mnt/cf
# Create GRUB directory hierarchy
mkdir /mnt/cf/boot && mkdir /mnt/cf/boot/grub
# Copy GRUB files to their place
cp stage1 stage2 e2fs_stage1_5 /mnt/cf/boot/grub/
cp menu.lst /mnt/cf/boot/grub/
# Unmount the CF
umount /mnt/fs
```

Now start grub by typing “grub” as a root user and write the MBR to the CF using grub shell as described in Listing 2.6. The command root defines in which disk and partition the root partition is located, e.g. where the boot loader is located. The disk and partitioning listing is independent from any operating systems and starts from zero.

Listing 2.6: Writing GRUB to the MBR

```
1 grub> root (hd0,1)
2 grub> setup (hd0)
3 grub> quit
```

The line 1 in Listing 2.6 defines that second partition on the first disk should contain the boot loader files. The setup command locates the required files on disk and writes the MBR to the disk. If the setup was successful, you should be able to boot GRUB in the target using the CF.

2.2 Selecting the Linux Distribution

The search for suitable Linux distribution was focused on open-source and embedded distributions. The considered distributions are listed in Table 2-2. The table shows that all the considered distributions have almost the same requirements and available features. All the operating systems have been released under license that guarantees them to be royalty-free and open-source.

Table 2-2: Feature comparison of considered Linux operating systems

Requirement	**EtLinux**	**eCos**	**Small Linux**
Version	1.1.3	2.0	0.8.1
Kernel	2.0.38	own	2.0
Processor	i386	i386	i386/i486
RAM memory	2MB	< 2MB	< 2MB
Disk space	2MB	few MB	≈ 4MB
Shell	X	-	X
TCP/IP networking	X	X	X
Multithreading	X	X	X
Serial	X	X	X
File system	X	-	X
Math emulation	X	X	X
Latest stable	Mar. 2005	May 2003	Dec. 2001
Latest update	Mar. 2005	May 2005	Dec. 2001
License	GPL	eCos License	LGPL

The Etlinux distribution is stated to been developed for small industrial computers, especially PC/104 boards [ETLinux, 2005]. The compatibility with PC/104 boards made the distribution a very potential option for our target. Unfortunately the ETLinux did not compile on our host computer and thus it had to be discarded.

Second tested distribution was eCos which is open source real-time operating system for embedded applications [eCos, 2006]. The eCos is stated not to be related to Linux but uses GNU C compiler and tools as most common Linux distributions. The eCos compiled and also booted in the host computer. Nevertheless, it did not boot on the target computer so it had to rejected also.

Last tested distribution was Small Linux which is targeted for old x86 systems with less than 2MB of memory [Small Linux, 2004]. The distribution is distributed both in binary and source format so it was quick to verify that it booted up in the target computer. The system seemed to work properly, fulfill all the set requirements for the operating system, and had still enough memory to run ZipMode application, so it was chosen as suitable distribution.

2.2.1 Small Linux Installation

The Small Linux is distributed in binary format using two floppy disks, namely boot and root disk. The package used for installation in this study is smalllinux081.tar.gz and it is available at [Small Linux, 2004]. To boot Small Linux using CF the contents of root and boot disks have to copied partially to the CF. This can be done by mounting the disk images into loop devices as shown in Listing 2.7. The root image contains directory called “Install” which contains a script that specifies which files are required for installation.

Listing 2.7: Installing the Small Linux binaries

```
# Mount root and boot disk images
mkdir boot root
```

```
mount -o loop smboot081 boot
mount -o loop smroot081 root
# Copy required files from root
cp -R bin etc lib sbin usr var root home proc mnt /mnt/hda2/
# Extract and copy files from boot disk
tar xzvf modules.tgz -C /mnt/hda2/lib/
tar xzvf extras.tgz -C /mnt/hda2/sbin/
cp linux system.map /mnt/hda2/
```

After copying all the files as shown in Listing 2.7, the Small Linux should be in the state that it is bootable with GRUB. The booting can be done directly from the GRUB shell, as shown in Listing 2.8, or by adding the boot command to menu.lst file.

Listing 2.8: Booting Small Linux using GRUB

```
root           (hd0,0)
kernel         /linux init=/bin/ash root=/dev/hda2
```

It might be useful to test the minimum requirements of your system in the host computer. The boot procedure is otherwise similar but you should give additional mem=2M parameter to the GRUB kernel command. It limits the available memory to 2MB and you can see if the operating system works or not without using target computer.

2.3 Compiling Applications for Small Linux

Small Linux is compiled to run binaries compiled against the glibc 2.0.7 library. This library can be downloaded e.g. as a rpm file: compat-glibc-5.2-2.0.7.1.i386.rpm. This can be then installed on 32-bit Debian environment by first converting it into Debian package, as shown in listing 2.9.

Listing 2.9: Installing glibc 2.0.7 library

```
alien compat-glibc-5.2-2.0.7.1.i386.rpm
dpkg -i compat-glibc_5.2-3_i386.deb
```

2.3.1 Hello World

To test that you can really compile and link against glibc 2.0.7 it is good to make a simple applications that only prints classical “Hello World” message on the screen. The C-code for “Hello world” is shown in listing 2.10.

Listing 2.10: Simple “Hello world” C-application

```
#include <stdio.h>
int main (void) {
  printf("Hello␣world!\n");
  return 0;
}
```

You can compile the “Hello world” applications using gcc compiler with nostdlib and nostdinc flags, as shown in listing 2.11. The application code is assumed to be in hw.c and the glibc 2.0.7 be installed in /usr/i386-glibc20-linux/. The output is a binary called hw.

Listing 2.11: Compiling the “Hello world” with gcc

```
gcc -s -nostdlib -nostdinc -I/usr/i386-glibc20-linux/include hw.o -
   L/usr/i386-glibc20-linux/lib /usr/i386-glibc20-linux/lib/crt1.o
   -lc -o hw
```

If your application compiled right, you should be able to see “Hello World” text on your target computer screen. Message like “FATAL: kernel too old” indicates that you failed to link the right libraries for your kernel.

The compiling command, shown in listing 2.11, is quite long for command line. For this reason it is more nice to write into a Makefile which compiles the application. A simple Makefile corresponding the listing 2.11 is shown in listing 2.12.

Listing 2.12: Compiling the “Hello world” with gcc using Makefile

```
SOURCES = hw.c
MAIN = hw.o
OBJECTS = $(MAIN) $(UTIL)
EXECUTABLE = hw

INCLUDEDIR=/usr/i386-glibc20-linux/include
LIBDIR=/usr/i386-glibc20-linux/lib

CC = gcc
LDFLAGS = -s -nostdlib
LDLIBS = -L$(LIBDIR) /usr/i386-glibc20-linux/lib/crt1.o -lc

.SUFFIXES:      .c
.c.o:
        $(CC) -c $(GCCFLAGS) $(CFLAGS) $(INCLUDE) $*.c
main:   $(MAIN) $(UTIL)
        $(CC) $(LDFLAGS) $(MAIN) $(UTIL) $(LDLIBS) -o $(EXECUTABLE)
depend: $(SOURCES) $(HEADERS)
        $(CC) -MM $(SOURCES) > Make.depend
hw.o : hw.c
```

It is possible to test the compiled programs in the host computer using chroot to change the operating environment. By changing to the root directory with command “chroot /mnt/hda2/” to the CF, you will be able to use the same binaries and libraries that are used in the target computer when running the compiled programs.

2.3.2 ZipMode Software

The provided ZipMode software is written with C language for MS-DOS operating system. This code is otherwise directly usable for Linux GNU C compiler except the serial interfaces used. Linux uses device interface to communicate with devices while in the MS-DOS they are usually accessed directly through low level functions, which are used in the Linux by the kernel.

The required changes are easy to implement by writing new versions from the serial interface functions. Also some variables, such as boolean, have to be redefined to be suitable for GNU C compiler. After these changes the ZipMode software can be compiled with identical Makefile as the “Hello World” application.

2.4 Cross-compiling Small Linux Kernel

The required packages to compile 2.0.0 kernel for Small Linux are

- gcc 2.7.2.3 (package: altgcc_2.7.2.3-2_i386.deb)
- glibc 2.0 (package: compat-glibc-5.2-2.0.7.1.i386.rpm)
- Small Linux 2.0.0 kernel (package: sm-0.7.2.src.tar.gz)
- Modified encaps, objdump, etc. binaries (package: smallcompile.tar.gz)

The GNU C compiler can be installed directly with Debian package and the glibc 2.0 can be installed by converting the rpm to Debian packet as described in Section 2.3. The kernel should be extracted and copied to /usr/src/ directory. The provided modified binary files should be replaced by existing ones in the /usr/bin and /usr/lib directories.

Some modifications have to be made to the provided compilation configuration because the kernel source is being compiled in the host computer. The Makefiles define compilation of some extra tools that are used to generate config files during the compilation. Problem is that some of these tools can not be compiled to run on the host computer but have to be compiled and run against glibc 2.0 libraries. The compilation of these extra tools have to be redefined so that they are compiled against the libraries which correspond the headers included.

The files can be compiled as shown in line 11 in Listing 2.13. Another change that has to be made is in the arch/i386/boot/compressed/Makefile. The CFLAGS have to changed to “CFLAGS = -nostdlib -I/usr/i386-glibc20-linux/include -L/usr/i386-glibc20-linux/lib -lc -O2 -DSTDC_HEADERS”. It defines that the given include and library definitions are used instead the default libraries. This way it can be ensured that the right libraries are linked against right headers.

To summarize, the files that have to be redefined for compilation are

- conmakehash (compiled with drivers/char/Makefile)
- misc.c (compiled with arch/i386/boot/compressed/Makefile)

Now the kernel can be compiled as shown in Listing 2.13. The above described CFLAGS redefinition of arch/i386/boot/compressed/Makefile has to be done before starting the compilation process.

Listing 2.13: Compiling Small Linux Kernel

```
# Link the kernel sources to /usr/include                                    1
cd /usr/include && rm -rf linux && rm -rf asm                                2
ln -s /usr/src/linux/include/linux linux                                     3
ln -s /usr/src/linux/include/asm-i386 asm                                    4
# Remove all previous compilation and config files                           5
make mrproper                                                                6
# Set the kernel configuration                                               7
make config                                                                  8
# Make dependencies (even though it fails)                                   9
make dep                                                                     10
# Recompile conmakehash and use it on the target environment                 11
cd drivers/char                                                              12
gcc -nostdlib -I/usr/src/linux/include -I/usr/i386-glibc20-linux/            13
    include -L/usr/i386-glibc20-linux/lib /usr/i386-glibc20-linux/
    lib/crt1.o -lc -O2 -c conmakehash conmakehash.c
```

```
cp conmakehash /mnt/hda2/ # Copy the conmakehash to CF
chroot /mnt/hda2 # Use the CF as root directory
./conmakehash cp437.uni > uni_hash.tbl
cp /mnt/hda2/uni_hash.tbl /usr/src/linux/drivers/char
# Compile dependencies again
make dep
# Compile the kernel zImage
make zImage
```

The compiled system should be around 284kBytes. The compressed kernel image can be found from arch/i386/boot/zImage. The new kernel can be installed to the target by simply replacing the old kernel on the CF.

2.4.1 Networking Interface

The network card in the PC/104 board is CS8900A and the kernel driver for this can be downloaded directly as binary from the Cirrus Logic web page. The kernel module can be loaded successfully to kernel with “insmod io=0x300” command. Unfortunately the network interface does not go up with ifconfig but gives “eth0: no network cable attached to configured media” and “SIOCSIFFLAGS: Resource temporarily unavailable” error messages.

The network card was tested with two Kontron MOPS/386A PC/104 boards and both gave the same error messages. This indicates that the problem is not probably a broken hardware. In the PC/104 main board manual this error is suggested to mean that the card does not detect signal from cable that is connected to a 100Mbit network. For this reason the box was also plugged to a 10Mbit network but the result was the same error.

The SIOCSIFFLAGS error message could also indicate that there is resource conflict with another device. For this reason a kernel with minimal device drivers was compiled, i.e. no floppy disk drivers, serial drivers, parallel port support, etc. Unfortunately the result was still the same error messages.

The CS8900 network driver sources are also publicly available for download. The driver source file is called cs89x0.c and the latest available version is v1.02 (11/26/96). The driver can be compiled against the kernel compiled previously in the Section 2.4, see Listing 2.14.

Listing 2.14: Compiling the CS8900 network card driver

```
gcc -c  -D__KERNEL__ -I/usr/src/linux/include -I/usr/src/linux/net/
    inet -Wall -Wstrict-prototypes -O2 -fomit-frame-pointer -DMODULE
     -DCONFIG_MODVERSIONS  cs89x0.c
```

The network card kernel module is easy to modify to ignore the network cable checking and to proceed like it would have been detected. After this modification the network card seems to work and the interface goes up with ifconfig without any error messages. Still there seems to be something wrong because no traffic is going in or out through the network card.

2.5 ZipMode Box Testing Results

The Small Linux operating system contains test script, located in /bin/tour, to demonstrate the system capabilities. It runs through most of the programs included with the distribution. The configured Small Linux system run successfully all of the tested programs except programs related to networking, which prompted that “undefined symbols found” while executing.

The upgraded ZipMode box was tested using real-time satellite data from an Astra satellite. The ZipMode box was fast enough to receive and process all the data without any exceptions. The processed measurements were saved successfully to CF, from where they could be sent forward when appropriate.

The ZipMode box was tested also with slower processor speeds than the default 40MHz. The ZipMode application software was able to run successfully still with 10MHz speed, but with 4MHz the speed was not anymore fast enough to read all the data from serial port. The event that requires most processing power is the data compression. The times required for compression with different processor speeds are shown in Table 2-3.

Table 2-3: ZipMode data compression speed

Processor speed	**Elapsed time**
4MHz	Failed
10MHz	0.741 seconds
25MHz	0.279 seconds
40MHz	0.174 seconds

The Small Linux kernel reports that it is allocating 1160kB of memory, leaving thus 924kB available for other purposes. Using a test program, which only reserves memory, it was possible without swap disk to reserve about 550kB of memory. This is much more than what is required by ZipMode application, which requires about 120kB of Random Access Memory (RAM) to run smoothly. Additionally it would be possible to activate part of the CF as a swap disk in case more memory would be needed.

Several small programs were coded with C and tested on the Small Linux. Most simple of them was “Hello world” application which just prints the “Hello World!” message to the screen. Another useful test program is serial echo program which reads data from given serial port and prints it on the screen. It can be used to record data for testing and to verify that data received is not corrupted. The actual ZipMode applications only uses these tested functionalities, i.e. serial connection and floating-point computation, so they are enough to verify system functionality for ZipMode application.

Based on the above results, Small Linux and the ZipMode application can be stated to work well with the given 386SX PC/104 computer. The only remained problem is the network card described in the Section 2.4.1. The fact that this problem was left unsolved is mostly due to the lack of time. The next logical steps for solving the problem with the network card would be

- Verify the functionality of network tools that were compiled for Small Linux network testing
- Verify the default settings of CS8900A (DOS based configuration program is provided by Cirrus Logic)
- Debug the network driver to see what it is doing (and what it is not)
- Contact the Cirruc Logic for user support

3 SATELLITE ORBIT DETERMINATION

The purpose of this chapter is to identify and analyze different possibilities for accurate satellite orbit determination, i.e. how to accurately determine the satellite ephemeris data. The motivation is that the satellite location information is directly affecting the satellite receivers' localization accuracy. By understanding the factors that contribute to the current high-end satellite POD, it might be possible to enhance the ZipMode system localization performance.

The Section 3.1 describes the different ways to determine satellite location at the orbit and the Section 3.2 describes how this orbit can be approximated using perturbation models. Last in the Section 3.3 a comparison of ZipMode and GPS system is conducted.

The satellite POD is important factor for ZipMode box localization, but not the only one. Also things such as

- Ionospheric and tropospheric delays
- Multipath reflections
- Receiver and transmitter noises (uplink, satellite, and downlink)
- Inaccurate uplink station localization
- Uplink measurement delays due to cabling
- Differences between used satellites (different delays)
- Earth model errors

have an affect on the accurate receiver localization. Some of the effects can be made smaller through calibration, e.g. uplink measurement delays, but can not be fully removed.

The satellite orbit determination is one of the most fundamental disciplines of satellite operations. Without knowledge of the satellite position, telecommand antenna can not be pointed to control the satellite. Especially for remote sensing satellites the relevancy of collected data is also inherently linked with POD.

The problem of orbit determination has existed since the launch of Sputnik which used signal from the Sputnik to characterize it's orbit parameters [Enge and Misra, 1999]. This problem is interesting also to the opposite direction which is utilized e.g. with Global Navigation Satellite System (GNSS). In this case the problem is to determine user location when satellite orbit is known.

The POD of satellites is difficult problem because the satellite is being affected by several disturbing forces. These forces have to be modelled or the localization measurement frequency have to be increased order to get submeter orbit determination accuracies.

An ideal circular satellite orbit is affected by following disturbing factors [Doornbos et al., 2003; Su, 2002]

- Earth's geopotential perturbations (Earth is not an ideal sphere)
- Solar attraction
- Lunar attraction
- Solar radiation pressure
- Atmospheric drag (especially on LEO)

- Attraction of major planets
- Solid Earth tides
- Ocean tides
- Ocean tide loading (deformation of crust due to ocean tides)
- Thermal re-radiation of the spacecraft
- Earth's albedo (Sun's radiation reflected by Earth)
- Space debris
- Propulsion, exhaust fumes/material

These factors have to be taken into account in any models developed. Some, like exhaust fumes, are more or less arbitrary and for that reason any model alone is not enough to provide accurate satellite ephemeris data.

3.1 Orbit Determination Principles

This section reviews the different satellite orbit determination methods in order to identify if they would be usable for ZipMode system performance enhancement. The focus is on the high-end solutions of each methods and their applicability for GEO localization. The advantages and disadvantages are summarized for each presented method.

The Table 3-1 presents different satellite orbit determination principles. The categorization can be done with several other ways, e.g. by dividing to range, Doppler and carrier phase methods based on the variable been measured. A more accurate division of methods is included in Appendix A.

Table 3-1: Satellite orbit determination methods classification

Determination type	Principle	Examples
One-way methods	Signal is received or sent by a satellite	GPS, DIODE, DORIS, NAVSAT
Two-way methods	Signal is received and sent by a satellite	SLR, microwave and radio wave ranging
Autonomous on-board methods	On-board sensors do the measurements for orbit determination	Horizon scanners, star tracker, gyroscopes, accelerometers and magnetometers

The one-way POD methods are identified by the fact that the signal travels distance from transmitter to receiver only once, see Figure 3-1. Typical these kind of methods are beacon based methods where the transmitter is assumed to be in a known location, e.g. satellite beacon based GPS or ground beacon based Doppler Orbiting and Radiopositioning Integrated by Satellite (DORIS), and the range is determined using TOF measurements. The one-way methods can be also Doppler and carrier phase measurements [Su, 2002]. For example GPS system can do Time of Flight (TOF), Doppler shift and carrier phase measurements. One other system using Doppler shift measurements is Navy Navigation Satellite System (NAVSAT) satellite system.

The two-way POD methods are methods where the signal travels two separate distances, typically arriving back to the same point of transmission, as shown in Figure 3-1. The two-way orbit determination can be based on TOF range or Doppler measurements. Typical TOF range measurement is the very accurate Satellite Laser Ranging (SLR) method.

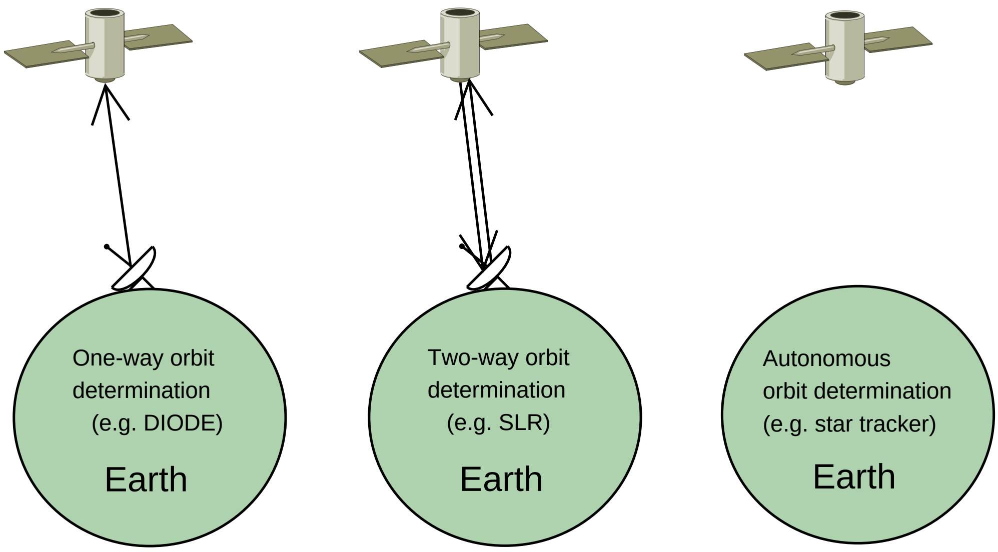

Figure 3-1: One-way, two-way and autonomous orbit determination

Autonomous on-board orbit determination uses on-board sensors to determine the satellite location, see Figure 3-1. The advantage is that it is independent and autonomous process. The achieved accuracy is less than e.g. with ranging but still acceptable for several missions. The relatively cheap price and low operation costs makes it also attractive for low cost missions.

3.1.1 One-way Methods

Since the world's first satellite, Sputnik, the on-ground methods for satellite orbit determination have been in use. In Sputnik's case the location of the satellite is determined in the Earth using the radio signal emitted from the satellite. The current high-end autonomous localization is done by various global navigation systems, like GPS and DORIS, which are capable to provide submeter orbit precision [Brunet et al., 1995].

The one-way methods can be further divided into TOF range, Doppler, and carrier phase measurements [Su, 2002]. The TOF measurement gives the distance to satellite when the measurement is multiplied with the speed of light. The Doppler measurements instead provides range-rate information based on the frequency shift. The carrier phase methods measure also range-rate by monitoring the speed of carrier phase change.

In orbit GPS receiver

The NAVigation Signal Timing And Ranging (NAVSTAR) GPS is currently the most widely used localization system on Earth. It has proven to be reliable, accurate and affordable solution for various localization tasks. For this reason it was finally tested and adopted for also satellite localization. For now the GPS receivers has been mainly used for tracking satellites on Low Earth Orbit (LEO), e.g. satellites such as TOPEX/POSEIDON (T/P) and CHAllenging Minisatellite Payload (CHAMP).

There has been one demonstration flight of GPS applicability to geosynchronous altitudes. It was done with United States satellite called Falcon Gold in 1997. The Falcon Gold successfully demonstrated that it was possible to get GPS signals useful for satellite localization also above the GPS constellation [Powell et al., 1999; Wu et al., 1992].

The principle of using GPS to locate GEO satellites is presented in Figure 3-2. The traditional GEO tracking is to do two-way ranging with one ground station, as e.g. with Digital Advanced Ranging with Transport-stream (DARTS) system. It is possible to increase this accuracy using several ground stations with long baseline, as shown in the middle of Figure 3-2. The GPS signals could be utilized with the same principles using spill-off GPS satellite signals, which are separated favorably by much longer baseline [Parkinson, 2002].

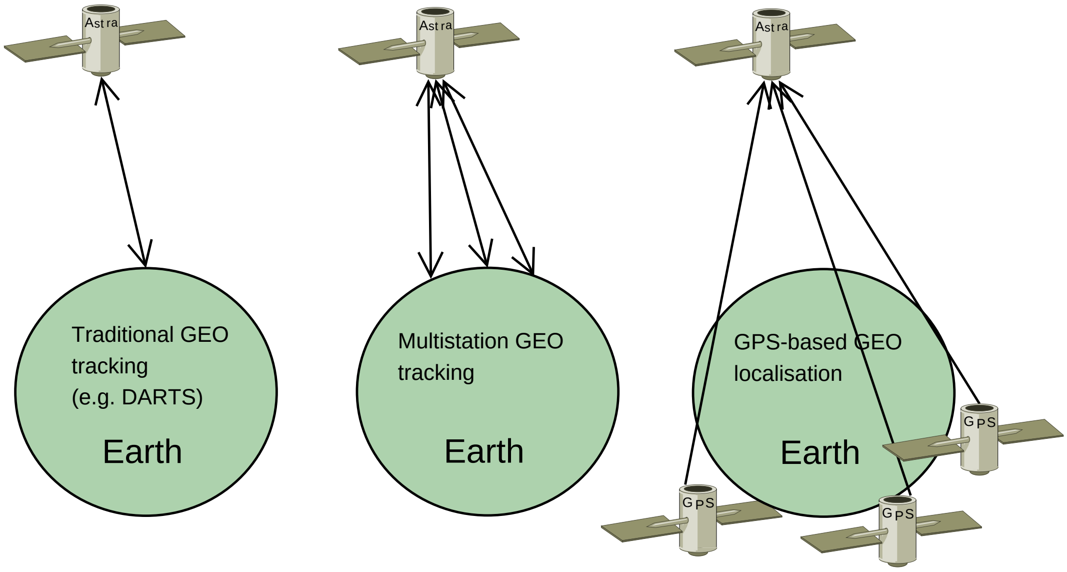

Figure 3-2: One and multi-station GEO ranging versus GPS localization

The GPS usage for orbit determination is based on the same principle as on the Earth. The GPS carrier phase can be used to detect range changes and pseudo-random noise modulated carrier signal, named P-code, gives measure of the absolute range [Lichten et al., 1988]. These two GPS data can be processed e.g. with Jet Propulsion Laboratory (JPL) orbit determination and baseline estimation software, named as GPS Inferred Positioning SYstem (GIPSY), to give sub-meter orbit determination accuracy at LEO. The advantages and disadvantages of using GPS are listed in Table 3-2.

Table 3-2: Advantages and disadvantages of in orbit GPS

Advantages	Disadvantages
• Maturity (widely used and flight proven technology) • Price (low-cost hardware) • Independent of weather conditions • Accuracy (submeter)	• Adds mass to the satellite (≈ 1kg) • Maturity for GEO satellites

Ground beacon systems

Satellite localization can be done also using ground based radio beacons. Most used such a system is currently DORIS designed by Centre National d'Etudes Spatiales (CNES), which is on-board e.g. SPOT2-5 and Envisat. DORIS system transmits dual frequency radio signal from a ground beacons that are distributed around the globe. On-board the satellite these two signals are translated into range-rate measurements [Doornbos et al., 2003]. The advantages and disadvantages of DORIS system are listed in Table 3-3. There has not been any satellites at geosynchronous orbits using DORIS.

Table 3-3: Advantages and disadvantages of DORIS

Advantages	Disadvantages
• Maturity (widely used and flight proven technology) • Accuracy (DORIS-DIODE ≈ 30cm) • Independent of weather conditions	• Adds mass to the satellite (≈20kg) • Applicability for GEO satellites not proven

3.1.2 Two-way Methods

The two-way orbit determination methods are based on measurement of electro-magnetic waves, usually operating in micro, radio or laser frequency. The two-way orbit determination can be based on TOF range or Doppler measurements. The TOF can be turned into range measurement by multiplying with speed of light. The Doppler measurements instead provide range-rate information based on the frequency shift.

The most accurate currently used two-way ranging method is SLR, which can achieve millimeter orbital determination precision [Doornbos et al., 2003]. The system currently used to localize ZipMode satellites is DARTS which is based on Round Trip Delay (RTD) ranging packet measurements [Harles et al., 2001, 2004]. The accuracy of DARTS orbit determination is around 100m which is several magnitudes lower than what SLR can provide.

Satellite Laser Ranging

In Satellite Laser Ranging the position of satellite is determined using round trip TOF measurements of short light pulses [Doornbos et al., 2003]. The positioned satellites have to be equipped with special Laser Retroreflector (LRR), also known as corner cube prisms, that are designed to reflect the light exactly back to the direction from which it came. The advantage of using special hardware is that it is possible to get millimeter positioning accuracy which is currently best positioning precision for Earth orbiting satellites [NASA, 1999]. The development of SLR orbit determination accuracy is shown in Figure 3-3.

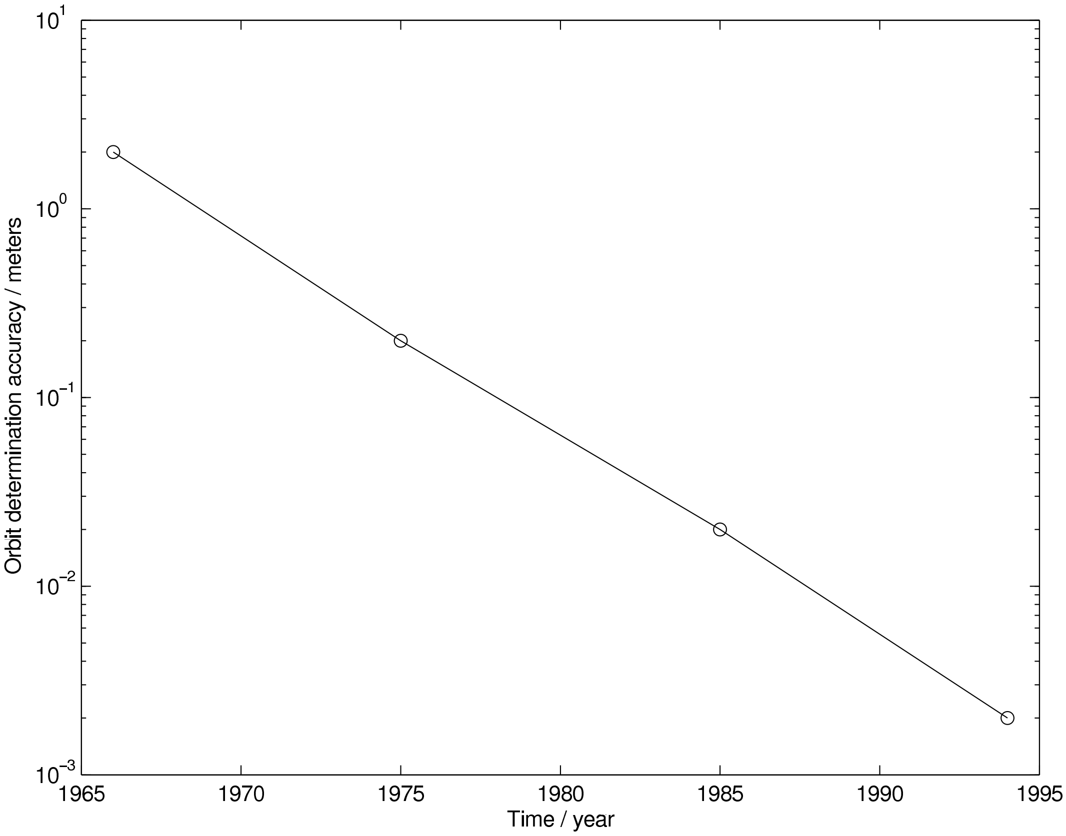

Figure 3-3: Development of SLR accuracy

The SLR is also applicable for GEO satellites through use of high power lasers to compensate the large distance. Etalon 1 and 2 passive satellites are example of high altitude, perigee at 19120km, satellites where SLR has been demonstrated. The laser ranging has been successfully used also to track the Moon during Apollo missions. Still at the moment SLR has not been used to locate GEO satellites [ILRS, 2006].

One advantage of SLR is that the laser link can be also used to provide Gbps range telecommunication link to the tracked satellite. There is effort by National Aeronautics and Space Administration (NASA) to develop an automated, low cost, sub-centimeter SLR system, called SLR 2000, with a laser communication capabilities version, named SLR 2000C [Degnan et al., 2004]. These two products have potential to bring the price of SLR low enough to make it the first choice for most of the satellite missions. The advantages and disadvantages of SLR are listed in Table 3-4.

Table 3-4: Advantages and disadvantages of SLR

Advantages	Disadvantages
• Accuracy (0.5-5cm) • Simplicity (passive retroreflector) • Mature flight proven technology	• Visibility requirements (weather, etc.) • Global coverage requires several stations • Adds mass to the satellite • Expensive to implement and operate

3.1.3 Autonomous On-board Methods

It is also possible to determine the satellite orbit location using only satellite on-board sensors, i.e. using magnetometers, Earth sensors, Sun sensors, star trackers and Inertial Measurement Unit (IMU)s. These kind of sensors usually lack behind in the accuracy but might be good option for low cost missions which do not require very accurate positioning and can operate autonomously. The achievable accuracies of autonomous system varies between one and 50km [Psiaki, 1999].

IMU is a system with integrated gyroscopes and accelerometer sensors. It can provide valuable information about the forces acting on the satellite but because it does not have any external sources of information it can provide only information relative to some previous position. For this reason IMU is usually accompanied by other absolute sensors, such as star trackers.

Horizon scanners and star trackers

The principle of Horizon scanners (Earth, Sun, Moon) and star trackers is the same. All of them use optical sensors to determine the location and attitude respect to objects which are in known locations. The advantage of these systems is that they are almost always usable and can operate autonomously. The major drawback is relatively poor accuracy compared e.g. with SLR.

The horizon scanner have been tested on-orbit e.g. on-board Technology for Autonomous Operational Survivability (TAOS) satellite payload. The flown instrument pack was Microcosm Autonomous Navigation System (MANS) and it contained Earth, Sun, and Moon sensors that were used to determine the satellite position and attitude. The achieved accuracy is around 100 meters [Hosken, 1995]. The advantages and disadvantages of magnetometers are shown in Table 3-5.

Table 3-5: Advantages and disadvantages of horizon scanners and star trackers

Advantages	Disadvantages
• Maturity (flight proven) • Autonomous • Price (medium range)	• Accuracy (from 100m to kilometers) • Mass (heavy optics, etc.)

Magnetometer

It possible to determine the satellite locations by measuring the variations of Earth's magnetic field. The achievable accuracy is between 8-125km [Psiaki, 1999]. One significant problem is the reliability of this method during space weather events because the shape of Earth's magnetic field changes and it should be taken into account in the system models for the localization to work. Especially in GEO the Earth's magnetic field can be pushed so low by space weather event that the satellite orbit is above the magnetic field. In this case the magnetometer data is not anymore usable for navigation. Unfortunately space weather modelling is not currently possible due to difficulties to predict Sun's behavior. Nevertheless, magnetometer is still potential option at LEO when it is combined with other sensors, e.g. with star trackers. The advantages and disadvantages of magnetometers are shown in Table 3-6.

Table 3-6: Advantages and disadvantages of Magnetometers

Advantages	**Disadvantages**
• Maturity (proven) • Autonomous • Simplicity (easy to implement) • Price (cheapest)	• Accuracy (8-125km) • Mass (sensor mass) • Not usable with GEO satellites (weak magnetic field)

3.1.4 Hybrid Methods

Because the fundamental importance of accurate and reliable satellite localization, most of the satellite mission do not rely only to one satellite localization system. Usually several orbit determination methods are used concurrently and the results are combined, e.g. with Kalman filter, into a usable position estimate. In most of the cases these systems are custom made for each mission, but also off-the-shelf orbit determination systems are available [Hosken, 1995].

For example T/P satellite uses five tracking systems for orbit determination; SLR, DORIS, GPS, Tracking and Data Relay Satellite System (TDRSS), and satellite altimeter measurements [Nerem et al., 1993]. Also the Jason-1 satellite, which is follow-on to T/P satellite, uses similar orbit determination hardware as T/P [Luthcke et al., 2003]. Both the T/P and Jason-1 have orbit accuracy of ≈ 2cm [Doornbos et al., 2003]. The advantages and disadvantages of this kind of combination are listed in the Table 3-7.

Table 3-7: Advantages and disadvantages of SLR, DORIS, GPS, and altimeter system

Advantages	**Disadvantages**
• Reliability (independent instruments) • Flight proven modules • Accuracy (≈ 1cm)	• Mass (several on-board instruments) • Price (complex and not autonomous)

The [Hosken, 1995] describes a low cost autonomous navigation system called MANS. It is capable to use GPS, gyroscope, accelerometer, Star, Earth, Moon, Sun sensor data. The MANS system was tested part of Space Test Experiment Platform (STEP) mission in 1994. It has been designed to provide position, speed and attitude information with accuracy of 400m (3σ), 0.4 $\frac{m}{s}$ and 0.05 degrees consecutively. The advantages and disadvantages of MANS system are listed in the Table 3-8.

Table 3-8: Advantages and disadvantages of MANS

Advantages	**Disadvantages**
• Reliability (independent instruments) • Autonomous • Flight proven module	• Accuracy (≈ 100m (1σ)) • Mass (several on-board instruments) • Price (not the cheapest option)

3.2 Precise Satellite Orbit Modelling

This section describes how satellite orbit can be modelled by taking account the perturbing forces acting on the satellite. Already simple modelling can enhance the ephemeris data significantly, e.g. as done for the ZipMode satellite orbits. The motivation of presented survey is to identify means to model satellite orbits and identify if they could be used for ZipMode satellite location.

The interesting parameters in orbit modelling are

- Satellite state vector (position, speed)

- Acting forces
- Measurement model parameters

The problem can be handled in three different ways [Hobbs and Bohn, 2005]

- Kinematic or geometric approach
- Dynamical approach
- Reduced dynamic approach

The kinematic approach is based on having accurate satellite location measurements and does not contain a dynamic satellite model. The modelling in this case is pretty straightforward orbit interpolation between measurement points.

Dynamic approach instead uses dynamic force models to take into account the acting forces. This approach is limited by the accuracy of used force models, such as geopotential and atmospheric drag models. Nevertheless it is most widely used approach for POD when accompanied with satellite tracking data.

Reduced dynamic approach combines the kinematic and dynamic approaches by using process noise model to filter dynamic model errors. This approach gives the best results still without increasing the complexity too much.

The most commonly modeled perturbation forces are listed in Table 3-9 for GEO satellite orbits [Su, 2002; Mervart, 1995]. The most significant forces for GEO are geopotential, Solar attraction, Lunar attraction and Solar radiation pressure, as shown in Figure 3-4. They should be at least accounted in order to keep the satellite localized with few kilometer accuracy.

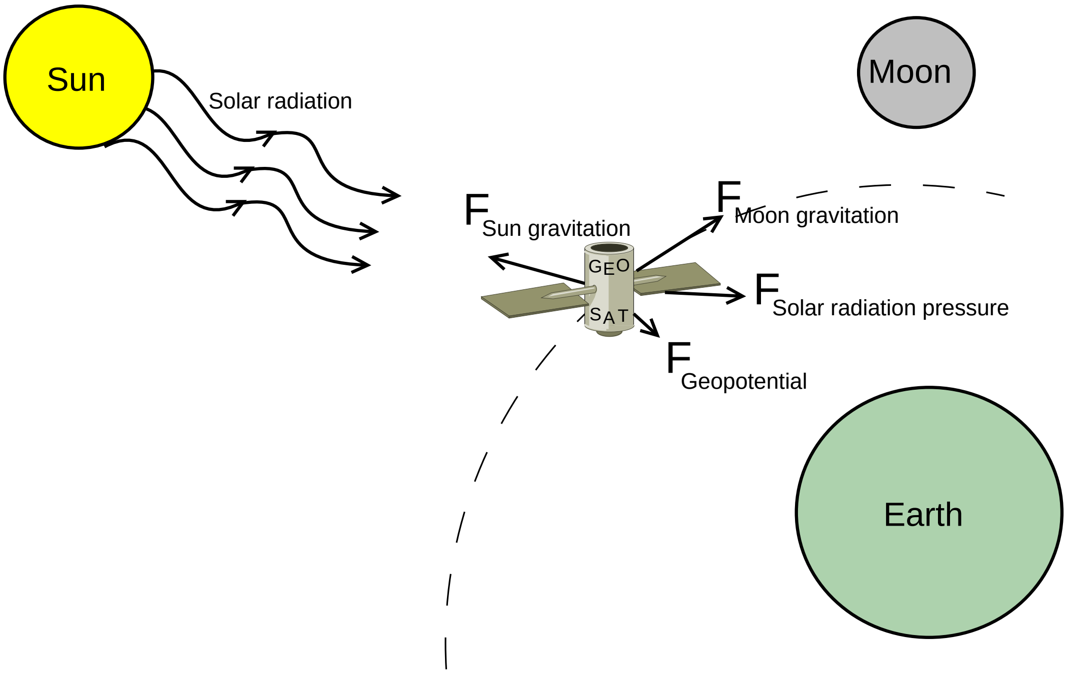

Figure 3-4: Most significant perturbing forces at GEO

3.2.1 Gravitational Perturbations

The satellites are orbiting Earth but are affected by more complex sum of different gravitational components than one ideal sphere. The gravitational forces can be expressed as a sum of different components, as shown in Equation 3.1 [Su, 2002].

Table 3-9: Satellite orbit perturbations for GEO satellites

Perturbation force	**Algorithm/Model**	**Error (m/day)**	**Explanation**
Earth's geopotential perturbation	Earth gravity model + potential field gradient	20000	Earth has non-spherical gravity field
Solar attraction	Point mass attraction	2263	Solar mass attracts the satellite
Lunar attraction	Point mass attraction	2311	Moon mass attracts the satellite
Solar radiation pressure	Sphere that absorbs all radiation	317	Solar radiation pressure moves the satellite
Solid Earth Tides	Deviations to Earth's gravity model coefficients	0.31	Deformations to Earth's figure and thus to gravity field due to other planetary bodies
Major planets attraction	Point mass attraction	0.02	Mass of major planets attracts the satellite
Thrust (exhaust gas, etc.)	Thrust + mass function	0	Exhaust and propulsion fumes cause perturbing force
Aerodynamic drag forces	Function of area, mass and air density	0	Earth's atmosphere creates a drag for the satellite
Ocean tides	Deviations to Earth's gravity model coefficients	0	Other planetary bodies affects oceans mass distribution and thus Earth's gravity field
Earth albedo radiation	Function of radiation pressure, area, mass, etc.	0	Reflection of Sun light produces force to the satellite

$$F_{total} = F_{GEO} + F_{st} + F_{ot} + F_n \qquad (3.1)$$

F_{GEO} = perturbations due to Earth's geopotential, i.e. gravitational attraction
F_{st} = perturbations due to Earth's solid tides
F_{ot} = perturbations due to Earth's ocean tides
F_n = perturbations due to other planetary bodies, i.e. Sun, moon and major planets

Main source for Earth's satellite orbital perturbations is the non-spheric form of Earth's body, i.e. the geopotential is not the same in the different parts of the orbit. The perturbing forces of Earth's gravitational attraction F_{GEO} can be calculated as a gradient of Earth's potential field. The Earth's potential can be presented as a point source which is extended to account non-spherical effects, as presented in [Su, 2002].

The solid tides F_{st} and ocean tides F_{ot} are due to the fact that Earth's mass distribution and shape are changing under the influence of other planetary bodies, especially the Moon and the Sun. The solid and ocean Earth tides can be taken into account as changes in the external geopotential, see [Rim and Schutz, 2002].

3.2.2 Solar Radiation Pressure

One of the significant perturbations in the GEO orbit is due to solar radiation pressure. It is the force generated by solar radiation, i.e. energy of electromagnetic waves radiated by the Sun. This energy is dependent on square of the distance to the Sun. The amount of energy radiated at Earth's distance from the Sun is about 1370 W/m^2.

The solar radiation pressure can be modelled as a function of surface area, distance to the Sun and solar energy coefficients [Su, 2002]. Already simple spheric assumption of the satellite shape can provide usable solution for orbit corrections.

3.2.3 *Other Perturbations*

Other perturbations such as Earth albedo radiation, satellite internal radiation and atmospheric drag, have negligible effect or are not important in GEO orbits [Su, 2002]. For example the Earth albedo radiation is the fraction of Sun radiation that is reflect off the Earth. At the LEO altitudes this has relevant impact to the orbit but at the GEO it can be neglected due to the huge distance. Exactly the same applies to atmospheric drag, it is relevant on LEO but not at the GEO due to the much lower particle density. LEO is between 200-1200km while GEO is around 35786km, i.e. about 30 times farther.

3.3 GPS and ZipMode System Comparison

In this section the ZipMode system is compared with the well known and more accurate GPS localization system in order to identify some possible ways to enhance the ZipMode localization precision. There is some some fundamental analogies between these two systems. Both of these systems use position of precisely located satellites to determine the location of the signal receivers and both of the systems use special timestamp packets for localization. Table 3-10 shows comparison between major ZipMode and GPS properties.

Table 3-10: Comparison of GPS and ZipMode systems

Property	**GPS**	**ZipMode**
Satellites used	Global satellite constellation of 24 satellites	Two or more colocated GEO satellites
Orbit altitude	≈ 20200km	≈ 36000km (GEO)
Localization principle	One way, from satellite to receiver	Two way, from ground via satellite to receiver
Number of satellites needed	3+1	2
Time needed for localization	seconds	hours
Type of positioning	Absolute positioning, RT and using post processing	Absolute positioning using post processing
Type of measurement	Signal TOF measurement	Signals TDOA measurement
Satellite orbit determination	Using IGS tracking network	DARTS two way ranging system

The major difference with these two systems is that the GPS does range measurements using accurately synchronized clocks when ZipMode box instead measures range differences using two separate satellite links, as shown in Figure 3-5. Also the calculation of absolute receiver position is done with the GPS in the receiver, when with the ZipMode system it is done on a separate centralized server. In practise this means that ZipMode can not provide any RT positioning.

The number of satellites required for GPS receiver localization is four; three for localization and one to remove timing variations [Walsh, 2003]. The TOF measurements to the GPS satellites need to be done only on a one time instance to determine the location. With ZipMode only two satellites are required for receiver localization because the movement of the satellites inside the colocation box is utilized. The TDOA measurements to the two satellites are done on a several different time instances that are separated usually by hours. These two different principles are shown in Figure 3-5.

For both of the systems the accurate satellite ephemeris data is the most important factor when doing accurate satellite receiver localization. The important orbital perturbation forces for GPS

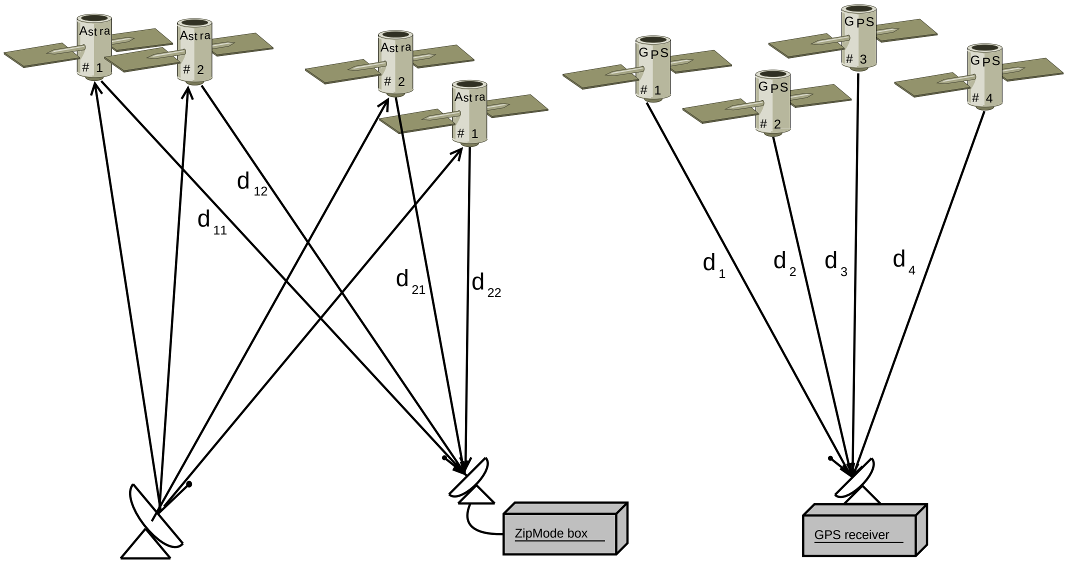

Figure 3-5: ZipMode and GPS receiver localization

and ZipMode satellite models are compared in Table 3-11. Also other properties affecting satellite receiver localization with these systems are considered in Table 3-12. The tables are based on results presented in [Su, 2002; Mervart, 1995; Walsh, 2003].

The orbit determination for ZipMode system satellites is done through system called DARTS [Harles et al., 2001]. DARTS system utilizes the payload transponders, due to higher available bandwidth, to measure RTD of two way ranging packets. The time difference is measured in the antenna output and in the antenna input using special DRR order to neglect non-deterministic delays. The reported 1σ timing accuracy of DARTS is as good as $\approx$ 14cm but other error contributions, such as atmosphere and delays, cause orbit determination accuracy to be tens of meters [Harles et al., 2001].

The GPS satellite orbits are determined accurately using IGS tracking network, which consist of about 230 tracking station equipped with high precision dual frequency, P-code geodetic GPS receivers [Su, 2002]. Data collected from the IGS tracking network is post-processed to give orbital ephemeris accuracy of 3-5cm. The same principle could be used for ZipMode satellite localization by broadcasting the timing packets to a reference receivers in a known locations. Based on accurate receiving time measurements, the satellite orbit position could be calculated as in the case of GPS satellites. Unfortunately the positioning would be more difficult and inaccurate due to the slow motion of the GEO satellites respect to the ground stations. Also the cost of building network of tens of reference receiver stations would be significant.

The ZipMode system uses also reference receivers to compensate some of the receiver location errors, such as orbital ephemeris disturbances and ionospheric delays. The principle is utilized widely also with GPS in the form of Differential GPS (DGPS). The idea is that by using a reference receiver in a known location, some of the error sources can be calibrated and removed from the receiver that is located in an unknown position [Walsh, 2003].

The GPS system has also capability to increase the localization accuracy using dual frequency receivers instead of single frequency receivers. The localization accuracy is more than double when using dual frequency GPS receivers. This might be applicable also for ZipMode system but the gain would be much smaller because the error caused by ionospherical delays are small compared to other error sources. Some changes would be also required to the ZipMode system setup which is not preferred.

Table 3-11: GPS satellite orbit modeling applicability for ZipMode satellites

Property	**GPS solution**	**Effect (m/day)**	**ZipMode applicability**	**Effect (m/day)**
Earth's geopotential	Modeled	10200	Modeled	20000
Effect of Moon and Sun	Modeled	3800	Modeled	4570
Solar radiation pressure	Modeled	200	Modeled (simple)	317
Solid Earth tides	Modeled	0.3	Neglected, insignificant at GEO	0.1
Effects of planets	Neglected, small effect	< 0.1	Also not significant at GEO	0.02
Ocean tides	Neglected, insignificant	< 0.01	Neglected, insignificant at GEO	< 0.04
Ocean tide loading	Neglected, hard to model	0.05	Neglected, insignificant	0.05
Albedo radiation pressure	Modeled	0.03	Neglected, insignificant at GEO	< 0.01
General relativistic effect due to Earth's gravity field	Neglected, small effect	< 0.03	Neglected, small effect	< 0.03
Atmospheric drag	Neglected, insignificant	0	Neglected, insignificant at GEO	0

Table 3-12: GPS localization error correction applicability for ZipMode

Property	**GPS solution**	**ZipMode applicability**
Clock accuracy	Time and frequency corrections	Frequency corrections
Ionospheric delays	Use of dual-frequency receivers (≈ 1.2-1.6GHz) and DGPS	Not significant factor with the used frequency band (12-16GHz)
Tropospheric delays	Modelling and DGPS	Not significant factor compared to other errors
Multipath	Measurement averaging, antenna design	Not significant, antenna has always LoS to satellite
Receiver noise	Signal noise, receiver resolution, etc. errors	Same type of errors as with GPS
Selective Availability (S/A)	Differential techniques	Signal not degraded on purpose
Geometrical Dilution of Precision (GDOP)	Satellite orbits calculated for good global coverage	Accounted when selecting appropriate satellites

One of the factors affecting receiver localization accuracy is so called GDOP. It is calculated as ratio between standard deviation and measurement accuracy [Walsh, 2003]. The GDOP can be used to measure the contribution of satellite constellation geometry to the positioning accuracy, and thus select the most suitable satellites for localization. The more satellites are spread around the horizon, the better the horizontal positioning but the weaker the vertical elevation accuracy, as shown in Figure 3-6.

The usage of multiple satellites is considered also for ZipMode system [Gross et al., 2006c]. The results imply that the initial localization accuracy is two decades better with three satellites than with two satellites. Generally, the more satellites can be used for localization, the better the accuracy.

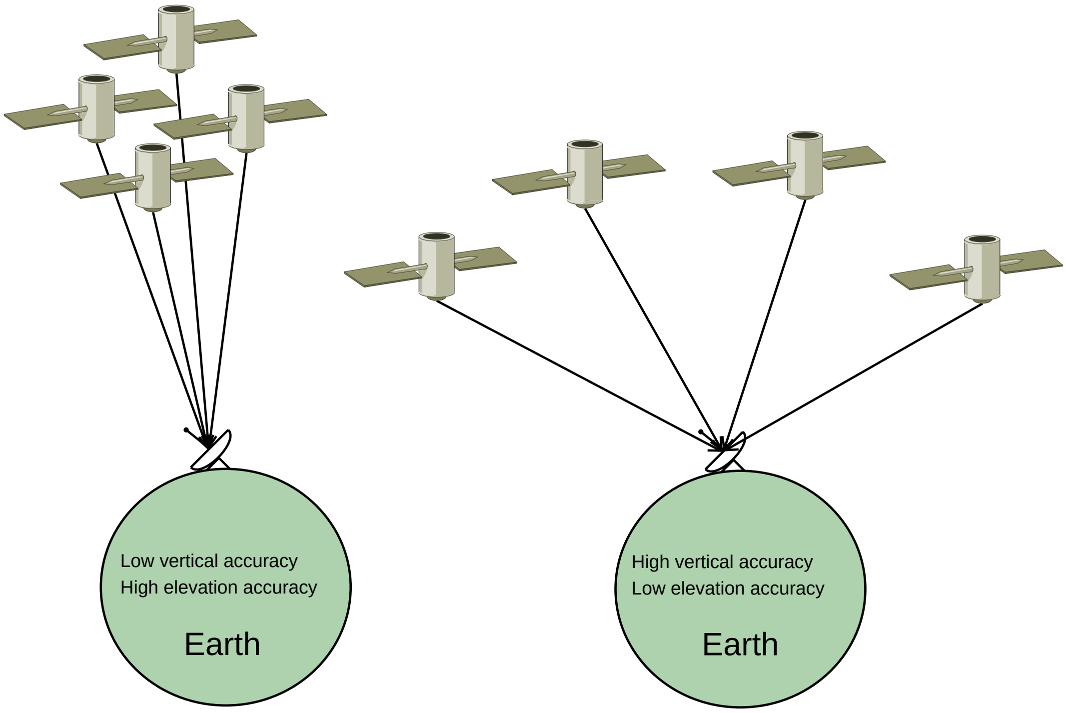

Figure 3-6: GDOP measures satellite formation efficiency

Based on the presented comparison it seems that the main reason for ZipMode system's lower location accuracy is the geometry and location of used satellites. Because of that the system localization capability is inherently lower than what e.g. GPS has. The system accuracy could be still increased by using the accurate models developed for the GPS system. For instance the orbit ephemeris accuracy could be increased by using more accurate models to account e.g. solar radiation pressure. This has direct effect to the localization accuracy.

4 CONCLUSIONS

This report presented an upgrade of a satellite receiver localization box, called ZipMode box. The ZipMode system is a DVB-S based satellite receiver localization system developed at prototype level at SES Astra SA. The goal of the ZipMode system is to develop a mass market localization system that can enable LBS, such as automatic regional program selection and targeted advertisement.

The first part of the report described the ZipMode box upgrade from a DOS based operating system environment into a Linux based environment. An open-source royalty free Linux operating system called Small Linux was selected to run on the ZipMode box. The Small Linux was demonstrated to run the ZipMode application code on a 386SX PC/104 computer equipped with 2MB of RAM. The upgraded ZipMode box was tested with real-time data received directly from an Astra satellite. The Small Linux was able to run the ZipMode application succesfully with 10MHz processor speed but not anymore with 4MHz. The ZipMode application required about 120kB of memory to run, leaving 430kB RAM available.

The latter part of the report reviewed the satellite orbit determination methods and existing models to account for forces perturbing the satellite orbits. State of the art satellite orbit determination is currently provided by GPS and SLR based systems, capable to few centimeter (RMS) orbital accuracy. On-orbit geopotential perturbations, solar radiation, Moon's attraction, and Sun's attraction are the most significant error sources for GEO satellites. They have to be accounted for order to achieve submeter orbital ephemeris precisions. The ZipMode system localization accuracy could be enhanced by combining accurate orbit models, existing already for GPS, with DARTS ranging measurements.

REFERENCES

Brunet, M., Auriol, A., Agnieray, P., and Nouel, F. [1995]. *DORIS precise orbit determination and localization system description and USOs performances*. In *Proceedings of the 49th IEEE International Frequency Control Symposium*, pages 122–132.

Degnan, J.J., Seas, A., Donovan, H., and Zagwodzki, T. [2004]. *SLR2000C: An autonomous satellite laser ranging and space-to-ground optical communications facility*. In *14th International Workshop on Laser Ranging*. San Fernando, Spain.

Doornbos, E., Naeije, M., and Scharroo, R. [2003]. *Final Report for GAMBLE WP 4: Orbit Determination and Satellite Tracking Workshop*. Final report for gamble wp 4, Delft University of Technology.

eCos [2006]. *Open source real-time operating system for embedded applications*. [Online; accessed 03-August-2006].
URL: `http://ecos.sourceware.org/`

Enge, P. and Misra, P. [1999]. *Special Issue on Global Positioning System*. In *Proceedings of the IEEE*, volume 87, pages 3–15. Stanford University, Stanford, USA.

ETLinux [2005]. *A Small Linux-based System for Industrial Computers*. [Online; accessed 03-August-2006].
URL: `http://www.prosa.it/etlinux`

Gross, M., Harles, G., Krier, G., and Nicolay, T. [2006a]. *A Localization System Using Geostationary Colocated Satellites For Location Based Services In Interactive TV*. In *Proceedings of Position Location and Navigation Symposium (PLANS)*. SES Astra SA.

Gross, M., Harles, G., and Krier, G. [2006b]. *ZipMode: Implementation of a Receiver Localization System using ASTRA Satellites*. In *Proceedings of 9th International Conference on Space Operations (SpaceOps)*. American Institute of Aeronautics and Astronautics (AIAA), Rome, Italy.

Gross, M., Harles, G., Krier, G., and Nicolay, T. [2006c]. *Optimization of a Receiver Localization System for SATMODE*. In *Ninth International Workshop on Signal Processing for Space Communications (SPSC 2006)*. ESTEC, Noordwijk, The Netherlands. Accepted for publication.

Harles, G., P.Siebert, Eberlein, E., Nowottne, H.J., Fritzsche, B., Haiduk, F., Knöchel, U., and Lindig, M. [2001]. *A novel ranging method using DVB-S transport stream packets*. In *Seventh International Workshop on Digital Signal Processing Techniques for Space Communications (DSP 2001)*. Sesimbra, Portugal.

Harles, G., Wouters, J., Fritzsche, B., and Haiduk, F. [2004]. *Operational Aspects of an Innovative, DVB-S based, Satellite Ranging Tool*. In *Proceedings of 7th International Conference on Space Operations (SpaceOps)*. Montreal, Canada.

Hobbs, D. and Bohn, P. [2005]. *Precise Orbit Determination for Low Earth Orbit Satellites*. In *Marie Curie Fellowships Annals*, volume 4.

Hosken, R. W. and Wertz, J.R. [1995]. *Microcosm Autonomous Navigation System on-Orbit Operation*. *Advances in the Astronautical Sciences*, 88:491.

ILRS [2006]. *The International Laser Ranging Service (ILRS)'s list of Satellites supported with SLR*. [Online; accessed 11-August-2006].
URL: `http://ilrs.gsfc.nasa.gov/satellite_missions/list_of_satellites/`

Karslioglu, M., Friedrich, J., and Urhan, H. [2005]. *A ground-based orbit determination for BILSAT*. In *Proceedings of 2nd International Conference on Recent Advances in Space Technologies, 2005. RAST 2005.*, pages 155–158.

Kim, S. and Chun, J. [2000]. *Bayesian bootstrap filtering for the LEO satellite orbit determination*. In *Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st*, volume 2, pages 1611–1615 vol.2.

Lichten, S.M., Bertiger, W.I., and Katsigris, E.C. [1988]. *A demonstration of Sub-meter GPS orbit determination and high precision user positioning*. In P.F. Wercinski, editor, *AAS/AIAA Astrodynamics Conference*, pages 8–16.

Luthcke, S.B., Zelensky, N.P., Rowlands, D.D., Lemoine, F.G., and Williams, T.A. [2003]. *The 1-Centimeter Orbit: Jason-1 Precision Orbit Determination Using GPS, SLR, DORIS, and Altimeter Data*. In *Marine Geodesy*, volume 26, pages 399–421. Taylor and Francis Inc.

Mervart, L. [1995]. *Ambiguity Resolution Techniques in Geodetic and Geodynamic Applications of the Global Positioning System*. Ph.D. thesis, University of Bern.

NASA [1999]. *Satellite Laser Ranging and Earth Science*. Technical report, National Aeronautics and Space Administration (NASA) Space Geodesy Program.

Nerem, R., Putney, B., Marshall, J., Lerch, F., Pavlis, E., Klosko, S., Luthcke, S., Patel, G., Williamson, R., and Zelensky, N. [1993]. *Expected orbit determination performance for the TOPEX/Poseidonmission*. *Geoscience and Remote Sensing, IEEE Transactions on*, 31(2):333–354. ISSN 0196-2892.

Parkinson, B.W. [2002]. *Satellite Navigation*. *Crosslink, The Aerospace Corporation magazine of advances in aerospace technology*, 2(2).

Powell, T.D., Martzen, P.D., Sedlacek, S.B., Chao, C.C., and Silva, R. [1999]. *GPS Signals in a Geosynchronous Transfer Orbit: "Falcon Gold" Data Processing*. Technical report, Air Force Academy Colorado Springs CO.

Psiaki, M.L. [1999]. *Autonomous LEO Orbit Determination From Magnetometer and Sun Sensor Data*. *Journal of Guidance, Control, and Dynamics*, 22(2):296–304.

Rim, H.J. and Schutz, B.E. [2002]. *Precise Orbit Determination (POD)*. Algorithm theoretical basis document, Center for Space Research, The University of Texas at Austin.

Small Linux [2004]. *Small linux for x86 systems with less than 2MB of RAM memory*. [Online; accessed 03-August-2006].
URL: `http://www.superant.com/smalllinux/`

Su, H. [2002]. *Precise Orbit Determination of Global Navigation Satellite System of Second Generation (GNSS-2)*. Ph.D. thesis, University of München.

Walsh, M.J. [2003]. *NAVSTAR Global Positioning System Surveying*. US Army Corps of Engineers, Washington, DC 20314-1000.

Wu, S., Yunck, T., Lichten, S., Haines, B., and Malla, R. [1992]. *GPS-based precise tracking of Earth satellites from very low togeosynchronous orbits*. In *Telesystems Conference, 1992. NTC-92., National*, pages 4/1–4/8.

APPENDIX A - ORBIT DETERMINATION COMPARISON

Table A-1: Satellite orbit determination methods

Reference	Method	Orbit	Accuracy	Verification	Algorithm
[Lichten et al., 1988]	Onground GPS receiver	20200km (GPS)	< 0.5m	Field tests	GIPSY
[Karslioglu et al., 2005]	Space-borne GPS receiver	686km (BILSAT)	< 1m	BILSAT GPS data	LSE
[Doornbos et al., 2003]	Radio beacons onground	790-1336km (e.g. T/P)	$\approx 30cm$	SPOT2-5, Envisat, etc.	DIODE
[Brunet et al., 1995]	Microwave beacons	830km (SPOT)	few cm	On-board e.g. SPOT 2	DORIS, PRARE
[Kim and Chun, 2000]	Magnetometer	$\approx$ 7000km	$<$ 80km	Simulation	BBF
[Psiaki, 1999]	Magnetometer + sun sensor	200-1000km (LEO)	$\approx$ 1700m (3σ)	Simulation	Batch filter
[Psiaki, 1999]	Magnetometer + star sensor	200-1000km (LEO)	$\approx$ 300m	Simulation	Batch filter
[Hosken, 1995]	Horizon scanners (MANS)	$\approx$ 555km	$\approx$ 400m	In-flight demo	Kalman filter
[Nerem et al., 1993]	SLR	1336km (T/P)	few cm	Inflight e.g. T/P	TOF measurements
[Harles et al., 2001]	DVB-S ranging	$\approx$ 35700km (GEO)	$\approx$ 100m	In commercial use	DARTS system