

Susana Maria Oliveira Ferreira

Seguimento de Plataformas Móveis

Dissertação do Mestrado em Engenharia Electrotécnica e de Computadores
Área de Especialização de Telecomunicações

Orientação: Prof. Doutora Benedita Malheiro



Instituto Superior de Engenharia do Porto
Departamento de Engenharia Electrotécnica
Rua Dr. António Bernardino de Almeida 431, 4200-072 Porto

Ano Lectivo: 2009-2010

Resumo

Este relatório apresenta o trabalho realizado no âmbito da unidade curricular de Tese/Dissertação do Mestrado em Engenharia Electrotécnica e de Computadores - área de especialização de Telecomunicações.

Pretende-se desenvolver um sistema distribuído de seguimento, no exterior, de plataformas móveis equipadas com receptores de baixo custo. O sistema deve, em tempo útil, realizar a aquisição, descodificação e tratamento dos dados emitidos pelo *Global Navigation Satellite System* (GNSS), das observações efectuadas pelo receptor e da informação proveniente do *European Geostationary Navigation Overlay System* (EGNOS). O objectivo é determinar, a partir deste conjunto de informação e para cada plataforma ligada, a posição em modo absoluto, as correcções diferenciais e, finalmente, a posição em modo diferencial.

Optou-se por receber as correcções diferenciais de área alargada do EGNOS através da Internet, permitindo, assim, que receptores sem capacidade de receber directamente informação do EGNOS possam também usufruir desta fonte de informação complementar.

As correcções diferenciais a aplicar às observações de cada receptor são geradas através do conceito de estação de referência virtual - *Virtual Reference Station* (VRS) - a partir da posição aproximada do receptor e das correcções de área alargada provenientes do EGNOS.

A determinação da posição em modo diferencial das plataformas móveis é efectuada segundo o conceito de *Inverted Differential Global Navigation Satellite System* (IDGNSS) e utilizando uma arquitectura do tipo Cliente-Servidor.

Por último, os resultados, que são armazenados numa base de dados, são disponibilizados ao utilizador através de uma aplicação *Web*. O utilizador pode, assim, efectuar o seguimento de qualquer plataforma móvel ligada ao sistema a partir de qualquer dispositivo com navegador e acesso à Internet.

Abstract

This report presents the project developed for the Thesis/Dissertation module of the Master Degree in Electrical Engineering and Computers, Telecommunications specialization area.

The goal is to develop a distributed system for the outdoor tracking of mobile platforms - rovers - equipped with low-cost Global Navigation Satellite Systems (GNSS) receivers. This is achieved through the acquisition, decoding and processing of GNSS and GNSS-related data, including the navigation message and observations collected by the receivers and the wide-area correction data from the European Geostationary Navigation Overlay System (EGNOS).

The system, first, determines, for each rover, the absolute position, the differential corrections and the differential position and, then, displays the rovers differential positions through a Web application.

The EGNOS wide-area corrections are acquired through the Internet to ensure that the access to this data source does not depend on the type of receiver or on the local receiving conditions.

The individual differential corrections are generated from the approximate position of each receiver and the EGNOS wide-area corrections according to the Virtual Reference Station (VRS) concept.

The individual differential position is determined according to the Inverted Differential Global Navigation Satellite System (IDGNSS) concept using a client-server architecture.

Finally, the differential positioning solutions, after being stored in a database, are displayed over a map through a Web application. This way, the user simply requires a device with a browser and an Internet connection to track any rover connected to this system.

Conteúdo

Conteúdo	i
Lista de Figuras	v
Lista de Tabelas	vii
Lista de Excertos de Código	ix
Lista de Equações	xi
Glossário	xv
Agradecimentos	xix
1 Introdução	1
1.1 Problema	1
1.2 Motivação	2
1.3 Objectivos	2
1.4 Estrutura do Relatório	3
2 Estado da Arte	5
2.1 Sistemas de Navegação e Posicionamento	5
2.1.1 Arquitectura	6
2.1.2 Fluxo da Informação	7
2.1.3 Posicionamento	7
2.1.4 Fontes de Erro	10
2.1.5 Sistemas de Navegação e Posicionamento Global	12
2.1.5.1 NAVigation Satellite Timing and Ranging Global Positioning System	12
2.1.5.1.1 Serviços	12
2.1.5.1.2 Segmento Espacial	13

2.1.5.1.3	Sinais	13
2.1.5.1.4	Mensagem de Navegação	14
2.1.5.1.5	Tipos de Observações	16
2.1.5.1.6	Mecanismos de Degradação Intencional	17
2.1.5.2	Globalnaya Navigational Sustenizavam Sistema	18
2.1.5.2.1	Segmento Espacial	18
2.1.5.2.2	Sinais	19
2.1.5.2.3	Mensagem de Navegação	19
2.1.5.3	Galileo	20
2.1.5.3.1	Serviços	20
2.1.5.3.2	Segmento Espacial	21
2.1.5.3.3	Sinais	22
2.1.5.3.4	Mensagem de Navegação	22
2.1.6	Sistemas de Navegação e Posicionamento Regional	22
2.1.6.1	Beidou/Compass	23
2.1.6.2	Quasi-Zenith Satellite System	24
2.2	Sistemas de Melhoramento	24
2.2.1	Sistemas de Melhoramento Suportados por Satélites	25
2.2.1.1	Wide-Area Augmentation System	27
2.2.1.2	European Geostationary Navigation Overlay System	27
2.2.1.3	System for Differential Correction and Monitoring	28
2.2.1.4	Multi-Functional Satellite Augmentation System	28
2.2.1.5	GPS Aided Geo Augmented Navigation	29
2.2.2	Sistemas de Melhoramento Suportados por Estações Terrestres	29
2.3	Sistemas de Navegação e Posicionamento Diferencial	30
2.3.1	Arquitetura e Princípios de Funcionamento	30
2.3.2	Sistemas Diferenciais Locais e Regionais	32
2.3.3	Sistemas Diferenciais de Área Alargada	32
2.4	Conclusão	33
3	Equipamento e Protocolos	35
3.1	Equipamento Utilizado	35
3.2	SiRF	37
3.2.1	Estrutura das Mensagens	37
3.2.2	Mensagens do Protocolo	37
3.3	RTCA DO-229	41
3.3.1	Estrutura das Mensagens	41
3.3.2	Mensagens da Especificação	41
3.4	SISNeT	45
3.5	RTCM SC-104	46

3.5.1	Estrutura e Tipos de Mensagens	46
3.6	Conclusão	49
4	Configuração do Ambiente de Desenvolvimento	51
4.1	Ambiente NetBeans IDE 6.9	51
4.2	MySQL	52
4.3	Servidor de Aplicações Tomcat	53
4.4	Aplicação SiRF	54
4.5	Aplicação ReadRawData	58
4.6	Aplicação SISNetClient/RTCA2RTCM	59
4.7	Conclusão	60
5	Desenvolvimento do Sistema	63
5.1	Arquitectura da Aplicação	63
5.2	Funcionalidades da Aplicação	64
5.2.1	Cliente SiRF Interface	65
5.2.1.1	Aquisição, Decodificação e Disponibilização de Dados	66
5.2.1.2	Posicionamento em Modo Absoluto	71
5.2.1.3	Interacção com o Serviço EGNOS VRS	82
5.2.1.4	Interacção com o Serviço IDGPS	84
5.2.2	Serviço EGNOS VRS	89
5.2.2.1	Aquisição, Decodificação e Tratamento das Cor- recções de Área Alargada do EGNOS	90
5.2.2.2	Correcções das Pseudodistâncias	95
5.2.2.3	Conversão de RTCA DO-229 para RTCM SC-104	101
5.2.2.4	Estação de Referência Virtual	105
5.2.3	Serviço IDGPS	108
5.2.3.1	Aquisição e Tratamento de Dados	110
5.2.3.2	Posicionamento em Modo Diferencial	112
5.2.3.3	Armazenamento de Dados	114
5.2.4	Aplicação Web Tracking of Mobile Platforms	117
5.3	Conclusão	122
6	Resultados dos Testes e Depuração do Código	123
6.1	Resultados dos Testes	123
6.2	Depuração do Código	128
6.3	Conclusão	128
7	Conclusão e Desenvolvimentos Futuros	129
7.1	Conclusão	129
7.2	Desenvolvimentos Futuros	130

Bibliografia

133

Lista de Figuras

2.1	Arquitectura dos GNSS	6
2.2	Posicionamento a uma, duas e três dimensões [1]	8
2.3	Solução de posicionamento com quatro satélites [1]	10
2.4	Logótipo do sistema NAVSTAR-GPS [2]	13
2.5	Órbitas do sistema NAVSTAR-GPS	14
2.6	Estrutura de uma mensagem de navegação [3]	15
2.7	Logótipo do sistema GLONASS [4]	18
2.8	Órbitas do sistema GLONASS [5]	19
2.9	Logótipo do sistema Galileo [6]	20
2.10	Órbitas do sistema Galileo [6]	21
2.11	Posicionamento no sistema Beidou [7]	23
2.12	Órbita do sistema QZSS [8]	24
2.13	Arquitectura do SBAS [9]	25
2.14	Cobertura dos diversos SBAS existentes [9]	26
2.15	Arquitectura do GBAS	30
2.16	Arquitectura do DGNS [3]	31
3.1	Receptor GNSS GM-210 da Holux [10]	35
3.2	Estrutura de um bloco RTCA DO-229 [11]	41
3.3	Relações entre as diversas mensagens RTCA DO-229 [11]	45
3.4	Cabeçalho de uma Mensagem RTCM SC-104 [12]	48
4.1	Desenvolvimento de aplicações com o NetBeans IDE	52
4.2	Modelação de bases de dados com o MySQL Workbench	53
4.3	Estrutura da aplicação <i>Sirf</i>	54
4.4	Lista das mensagens SiRF recebidas	55
4.5	Mensagem SiRF 4 - <i>Measured Tracker Data</i>	56
4.6	Mensagem SiRF 10 - <i>Error ID</i>	56
4.7	Mensagens SiRF que podem ser solicitadas	57

4.8	Mensagem SiRF 144 - <i>Poll Clock Status</i>	57
4.9	<code>ReadRawData</code> : execução	59
4.10	SISNeTClient/RTCA2RTCM: autenticação no serviço SISNeT	60
4.11	SISNeTClient/RTCA2RTCM: recepção e decodificação de dados EGNOS 61	
5.1	Sistema de seguimento de plataformas móveis: arquitectura	64
5.2	Sistema de seguimento de plataformas móveis: diagrama funcional . .	65
5.3	Mensagem SiRF 28 - <i>Navigation Library Measurement Data</i>	71
5.4	Mensagem SiRF 15 - <i>Ephemeris Data</i>	72
5.5	Mensagem SiRF 30 - <i>Navigation Library SV State Data</i>	73
5.6	Representação de coordenadas ECEF e ENU	76
5.7	SiRF Interface: resultados do algoritmo <code>PosAbs</code>	82
5.8	SiRF Interface: interacção com o EGNOS VRS Server	86
5.9	SiRF Interface: envio de dados IDGPS Server	89
5.10	SISNeT: mensagens utilizadas	91
5.11	EGNOS VRS Server: decodificação das mensagens RTCA DO-229 . .	96
5.12	EGNOS VRS Server: interacção com o SISNeT	97
5.13	IPP em relação a um receptor e a um satélite [13]	100
5.14	Pontos da grelha ionosfera e respectivas faixas [13]	101
5.15	EGNOS VRS Server: atendimento dos clientes SiRF Interface	109
5.16	IDGPS Server: atendimento dos clientes SiRF Interface	112
5.17	IDGPS Server: resultados do método <code>updatePVT()</code>	114
5.18	Modelo da base de dados <code>trackingDB</code>	115
5.19	Página <i>Web</i> da aplicação <i>Web Tracking of Mobile Platforms</i>	121
5.20	Seguimento de um <i>rover</i>	121
6.1	Solução PVT do receptor Holux GM-210	124
6.2	Solução PVT do <code>PosAbs</code>	124
6.3	Solução PVT: receptor <i>vs</i> <code>PosAbs</code>	125
6.4	Solução PVT: distribuição dos valores do receptor	125
6.5	Solução PVT: distribuição dos valores do <code>PosAbs</code>	126
6.6	IDGPS Server em execução	127
6.7	Solução PVT do <code>PosDif</code>	127

Lista de Tabelas

3.1	Estrutura geral das mensagens SiRF	37
3.2	Mensagens SiRF de saída solicitáveis	39
3.3	Mensagens RTCA DO-229	44
3.4	Mensagens RTCM SC-104	47
3.5	MT1 RTCM SC-104 - <i>Differential GPS Corrections</i> [12]	48
3.6	MT3 RTCM SC-104 - <i>GPS Reference Station Parameters</i> [12]	49
5.1	SiRF MT28 - <i>Navigation Library Measurement Data</i>	66
5.2	SiRF MT15 - <i>Ephemeris Data</i>	69
5.3	SiRF MT30 - <i>Navigation Library SV State Data</i>	70

Lista de Excertos de Código

5.1	Excerto da classe <code>Mensagem28</code> , subclasse de <code>GPsmensagem</code>	68
5.2	Excerto da classe <code>Mensagem15</code> , subclasse de <code>GPsmensagem</code>	70
5.3	Excerto do método <code>inicializaPVT()</code>	81
5.4	Método construtor da classe <code>connectEGNOS</code>	83
5.5	<i>Thread</i> <code>SendData</code> , responsável pelo envio de dados entre o <code>SiRF Interface</code> e o <code>EGNOS VRS Server</code>	84
5.6	<i>Thread</i> <code>ReceiveData</code> , responsável pela recepção de dados provenientes do <code>EGNOS VRS Server</code>	85
5.7	Método construtor da classe <code>connectIDGPS</code>	87
5.8	<i>Thread</i> <code>SendData</code> , responsável pelo envio de mensagens entre o <code>SiRF Interface</code> e o <code>IDGPS Server</code>	87
5.9	<i>Thread</i> <code>ReceiveData</code> , responsável pela recepção das respostas do <code>IDGPS Server</code>	88
5.10	Método construtor da classe principal <code>egnosServer</code>	90
5.11	Método <code>processAnswer()</code> da classe <code>SISNetClient</code> que escuta a chegada de mensagens provenientes do <code>SISNet</code>	92
5.12	Classe <code>DataBlock</code> , que decodifica a informação recebida do serviço <code>SISNet</code> no formato <code>RTCA DO-229</code>	93
5.13	Chegada de uma mensagem <code>EGNOS</code> e invocação do método <code>generateRTCM()</code>	102
5.14	Método <code>generateRTCM()</code> que cria as mensagens <code>RTCM SC-104</code> com as correcções diferenciais	103
5.15	Método <code>makeRtcmMsg()</code> que cria as mensagens <code>RTCM SC-104 MT1</code> e <code>MT3</code>	104
5.16	Método construtor da classe <code>VRSServer</code> e criação das <i>threads</i> <code>HandleClient</code> de atendimento dos clientes <code>SiRF Interface</code>	106
5.17	Escuta e processamento das mensagens na classe <code>HandleClient</code> do <code>EGNOS VRS Server</code>	107
5.18	Método <code>generateClientRTCM()</code> da classe <code>SISNetClient</code>	108

5.19	Método construtor da classe principal <code>IDGPServer</code> e método <code>execute()</code> que escuta a chegada de clientes <code>SiRF Interface</code>	110
5.20	<code>IDGPS Server</code> : processamento das mensagens dos clientes <code>SiRF Interface</code>	111
5.21	<i>Setters</i> e <i>getters</i> do <i>bean</i> utilizado para interagir com a base de dados <code>TrackingBean</code>	117
5.22	Método <code>toDataBase()</code> da classe <code>clientPanel</code> , que invoca os <i>setters</i> do <code>TrackingBean</code>	117
5.23	Método <code>updateDBRecord()</code> da classe <code>TrackingBean</code> , que estabelece a ligação com a base de dados e preenche as tabelas	118
5.24	<i>Servlet</i> <code>MapServlet</code> e o seu método <code>doGet()</code> que processa os pedidos dos clientes	119
5.25	Escuta dos botões que representam os <i>rovers</i> feita pelo objecto <code>request</code>	119
5.26	<i>Script</i> de JavaScript que faz a invocação da API do Google Maps .	120

Lista de Equações

2.1: Distância teórica entre satélite e receptor	7
2.2: Erro de medida da distância entre satélite e receptor	8
2.3: Pseudodistância corrigida	9
2.4: Posição do receptor	9
5.1: Pseudodistância bruta observada pelo receptor	67
5.2: Frequência da portadora observada pelo receptor	67
5.3: Fase da portadora observada pelo receptor	67
5.4: <i>GPS Software Time</i> observado pelo receptor	67
5.5: Tempo de transmissão do satélite	67
5.6: Tempo de recepção do receptor	68
5.7: Posição ECEF ajustada do satélite	71
5.8: Diferença dos tempos de transmissão e das efemérides do satélite	72
5.9: Semi-eixo maior	72
5.10: Movimento médio	73
5.11: Movimento médio corrigido	73
5.12: Anomalia média	73
5.13: Anomalia excêntrica	74
5.14: Anomalia verdadeira	74
5.15: Argumento da latitude	74
5.16: Argumento da correcção da latitude	74
5.17: Argumento da correcção do raio	74
5.18: Argumento da correcção da inclinação	74
5.19: Latitude corrigida	75

5.20: Raio corrigido	75
5.21: Inclinação corrigida.....	75
5.22: Coordenadas orbitais X e Y do satélite.....	75
5.23: Longitude corrigida do nodo ascendente.....	75
5.24: Coordenadas ECEF do satélite	75
5.25: Diferença entre o tempo de transmissão e do relógio do satélite.....	76
5.26: Termo de correcção relativista	76
5.27: Desvio de relógio do satélite	77
5.28: Factor de obliquidade da ionosfera.....	77
5.29: Ângulo central do receptor com a projecção terrestre do IPP.....	77
5.30: Latitude geodésica da projecção terrestre do IPP	77
5.31: Longitude geodésica da projecção terrestre do IPP.....	77
5.32: Latitude geomagnética da projecção terrestre do IPP	78
5.33: Tempo local.....	78
5.34: Parâmetro da ionosfera PER	78
5.35: Parâmetro da ionosfera AMP	78
5.36: Parâmetro da ionosfera x	78
5.37: Atraso ionosférico	78
5.38: Atraso troposférico	79
5.39: Pseudodistância corrigida em modo absoluto.....	79
5.40: Algoritmo PosAbs: Distância do receptor ao satélite.....	80
5.41: Algoritmo PosAbs: Componentes da variação da distância	80
5.42: Correcção rápida da $range-rate$	97
5.43: Correcção rápida da distância total	98
5.44: Correcção de longo prazo da posição do satélite	98
5.45: Correcção de longo prazo do relógio do satélite.....	99
5.46: Componente do relógio do satélite de correcção da distância.....	99
5.47: Componente troposférica de correcção da distância.....	99
5.48: Componente ionosfera de correcção da distância	101
5.49: VRS: Correcção da pseudodistância.....	101
5.50: <i>Modified Z-Count</i>	107
5.51: Correcção diferencial da pseudodistância.....	113

5.52: Pseudodistância ajustada.....	113
5.53: Pseudodistância corrigida em modo diferencial.....	113

Glossário

Abreviatura	Descrição	Definição
GNSS	<i>Global Satellite Navigation System</i>	página 1
PVT	Posição, Velocidade e Tempo	página 1
ISEPSat	Laboratório de Comunicações por Satélite do Instituto Superior de Engenharia do Porto	página 2
ISEP	Instituto Superior de Engenharia do Porto	página 2
IDGNSS	<i>Inverted Differential GNSS</i>	página 2
VRS	<i>Virtual Reference Station</i>	página 2
NAVSTAR-GPS	<i>Navigation Satellite Timing and Ranging Global Positioning System</i>	página 5
GLONASS	<i>Globalnaya Navigatsionnaya Sputnikovaya Sistema</i>	página 5
QZSS	<i>Quasi-Zenith Satellite System</i>	página 5
SV	<i>Space Vehicle</i>	página 6
DGNSS	<i>Differential Global Navigation Satellite System</i>	página 10
EUA	Estados Unidos da América	página 12
JPO	<i>Joint Program Office</i>	página 12
DoD	<i>Department of Defense</i>	página 12
SPS	<i>Joint Program Office</i>	página 12
PPS	<i>Precise Positioning Service</i>	página 12
SA	<i>Selective Availability</i>	página 13
h	hora	página 13
min	minuto	página 13
MCS	<i>Master Control Station</i>	página 13
CDMA	<i>Code Division Multiple Access</i>	página 13
C/A	<i>Coarse/Aquisition</i>	página 13
P(Y)	<i>Encrypted Precise</i>	página 14
M	<i>Military</i>	página 15
d	dia	página 14
CM	<i>Civil Moderate</i>	página 15
PRN	<i>Pseudorandom Noise</i>	página 14
b	bit	página 14

Abreviatura	Descrição	Definição
TLM	<i>Telemetry Word</i>	página 16
HOW	<i>Handover Word</i>	página 16
A-S	<i>Anti-Spoofing</i>	página 17
URSS	União das Repúblicas Socialistas Soviéticas	página 18
TTC	<i>Telemetry, Tracking and Control</i>	página 18
FDMA	<i>Frequency Division Multiple Access</i>	página 19
ESA	<i>European Space Agency</i>	página 20
OS	<i>Open Service</i>	página 20
SOL	<i>Safety-Of-Life</i>	página 20
PRS	<i>Public Regulated Service</i>	página 20
CS	<i>Commercial Service</i>	página 20
SAR	<i>Search-And-Rescue</i>	página 20
GCS	<i>Ground Control Segment</i>	página 21
GMS	<i>Ground Mission Segment</i>	página 21
GSS	<i>Galileo Sensor Station</i>	página 21
RNSS	<i>Regional Satellite Navigation System</i>	página 22
SBAS	<i>Space-Based Augmentation System</i>	página 25
GBAS	<i>Ground-Based Augmentation System</i>	página 25
WAAS	<i>Wide-Area Augmentation System</i>	página 26
EGNOS	<i>European Geostationary Navigation Overlay System</i>	página 26
SDCM	<i>System for Differential Correction and Monitoring</i>	página 26
MSAS	<i>Multi-Functional Satellite Augmentation System</i>	página 26
GAGAN	<i>GPS Aided Geo Augmented Navigation</i>	página 26
RTCA	<i>Radio Technical Commission for Aeronautics</i>	página 26
FAA	<i>Federal Aviation Administration</i>	página 27
WMS	<i>Wide-Area Master Station</i>	página 27
WRS	<i>Wide-Area Reference Station</i>	página 27
RIMS	<i>Receiver Integrity Monitoring Station</i>	página 27
MCC	<i>Master Control Center</i>	página 27
SIS	<i>Signal in Space</i>	página 27
SISNeT	<i>Signal in Space over Internet</i>	página 28
IRNSS	<i>Indian Regional Navigational Satellite System</i>	página 29
LAAS	<i>Local-Area Augmentation System</i>	página 29
VHF	<i>Very High Frequency</i>	página 29
RTCM	<i>Radio Technical Commission for Maritime Services</i>	página 29
MT	Mensagem do Tipo	página 47
RTK	<i>Real Time Kinematics</i>	página 32
LADGNSS	<i>Local-Area Differential GNSS</i>	página 32
RADGNSS	<i>Regional-Area Differential GNSS</i>	página 32
WADGNSS	<i>Wide-Area Differential GNSS</i>	página 32
B	byte	página 37
DOP	<i>Dilution of Precision</i>	página 38
TOW	<i>Time of Week</i>	página 39

Abreviatura	Descrição	Definição
HDOP	<i>Horizontal DOP</i>	página 39
MOPS	<i>Minimum Operational Performance Standards for Global Positioning System/Wide Area Augmentation System Airborne Equipment</i>	página 41
IODP	<i>Issue Of Data: PRN</i>	página 42
UDRE	<i>User Differential Range Error</i>	página 42
IODF	<i>Issue Of Data: Fast Correction</i>	página 42
UDREI	<i>User Differential Range Estimate Indicator</i>	página 42
URA	<i>User Range Accuracy</i>	página 42
NPA	<i>Non Precision Approach</i>	página 43
IODI	<i>Issue Of Data: Ionospheric</i>	página 43
IODS	<i>Issue Of Data: Service</i>	página 43
IOD	<i>Issue Of Data</i>	página 44
SINCA	<i>SiSNeT Compression Algorithm</i>	página 46
ECEF	<i>Earth-Centered, Earth-Fixed</i>	página 48
IDE	<i>Integrated Development Environment</i>	página 51
SE	<i>Standard Edition</i>	página 51
JDK	<i>Java Development Kit</i>	página 51
JRE	<i>Java Runtime Environment</i>	página 51
TCP	<i>Transfer Control Protocol</i>	página 63
IDGPS	<i>Inverted Differential Global Positioning System</i>	página 64
ENU	<i>East-North-Up</i>	página 76
IPP	<i>Ionospheric Pierce Point</i>	página 77
WGS 84	<i>World Geodetic System 1984</i>	página 77
IDT	Instituto de Desenvolvimento Tecnológico	página 78
IGP	<i>Ionospheric Grid Points</i>	página 94
MZC	<i>Modified Z-Count</i>	página 107
SQL	<i>Structured Query Language</i>	página 116
AJAX	<i>Asynchronous JavaScript And XML</i>	página 130

Agradecimentos

Gostaria de expressar o meu agradecimento a todas as pessoas que, de alguma forma, contribuíram para a realização deste trabalho de Dissertação e me auxiliaram nos momentos de dificuldade.

Não quero deixar de prestar o meu profundo agradecimento ao Prof. Doutor Manuel Gonçalves Soares, antigo docente do ISEP, por me ter proposto este trabalho quando ainda estava em funções na instituição e por todo o conhecimento que me transmitiu na área da navegação e comunicações por satélite.

Do mesmo modo, quero agradecer à minha orientadora, Prof. Doutora Benedita Malheiro, por todo o apoio científico e pedagógico que me prestou ao longo deste trabalho, pela motivação que me transmitiu e pela cedência de equipamentos, informação e instalações, sem os quais este trabalho não conseguiria ser realizado.

Por fim, quero agradecer à minha família e amigos pelo apoio incondicional a cada momento e pela confiança que sempre demonstraram em mim.

Capítulo 1

Introdução

Actualmente, a determinação da localização precisa de pessoas e bens tornou-se uma necessidade básica. Este serviço é fornecido conjuntamente pelos Sistemas Globais de Navegação por Satélite ou Global Satellite Navigation System (GNSS), que são redes globais de satélites capazes de transmitir sinais de rádio contendo informação que permite determinar a localização de um receptor apropriado em qualquer ponto do planeta, e pelos sistemas complementares de melhoria da precisão.

1.1 Problema

O posicionamento baseado em satélites consiste na determinação de posição, velocidade e tempo (PVT) em qualquer local (terra, mar, ar ou espaço). No entanto, este tipo de posicionamento, designado posicionamento em modo absoluto, está sujeito a erros provenientes de um conjunto variado de origens, o que torna estes sistemas inadequados para algumas situações como, por exemplo, aviação civil. Existem métodos e tecnologias, nomeadamente o posicionamento em modo diferencial, que procuram reduzir esses erros, tornando o posicionamento baseado em satélites numa tecnologia de aplicação mais ampla.

Este trabalho pretende desenvolver um sistema no domínio do posicionamento diferencial, *i.e.*, que determine e aplique as correcções diferenciais que se aplicam às posições das plataformas móveis ligadas.

1.2 Motivação

Este projecto, que foi desenvolvido no Laboratório de Comunicações por Satélite (ISEPSat) do Instituto Superior de Engenharia do Porto (ISEP), constitui um sistema de seguimento de plataformas móveis a aplicar ao contexto dos transportes.

Este sistema deverá interagir com receptores GNSS de baixa gama, obter os dados brutos e aplicar, num nó central, as correcções diferenciais, implementando assim o conceito de *Inverted Differential GNSS* (IDGNSS). Os dados resultantes, uma vez armazenados numa base de dados, poderão finalmente ser acedidos externamente através de uma aplicação *Web* de visualização gráfica da informação de posição, *i.e.*, realizar o seguimento de plataformas móveis dotadas de receptores GNSS de baixo custo.

O equipamento utilizado no desenvolvimento deste trabalho foi constituído por receptores GNSS Holux, modelo GM-210, que têm um custo unitário de 36 €.

1.3 Objectivos

Pretende-se desenvolver um sistema que determine, através do conceito IDGNSS, a posição precisa de um conjunto de plataformas móveis. Para tal, o sistema tem de adquirir as correcções de área alargada do EGNOS, determinar as correcções diferenciais, aplicá-las aos dados recebidos de receptores SiRF instalados em plataformas móveis e disponibilizá-las através de uma aplicação *Web*. O sistema deverá ser realizado quer implementando novos módulos dedicados, quer reutilizando módulos de *software* pré-existentes.

Assim e após estudados os módulos a reutilizar, a aplicação desenvolvida deverá ser capaz de:

- Recolher os dados recebidos pelo receptor GNSS utilizando o protocolo SiRF, nomeadamente as medidas brutas, as efemérides dos satélites e outros dados relevantes;
- Com base nesta informação, calcular a posição dos satélites visíveis e a posição do receptor em modo absoluto;
- Adquirir, através da Internet, as correcções de área alargada do EGNOS, decodificá-las do formato RTCA DO-229, gerar as correcções diferenciais específicas para cada receptor através da técnica *Virtual Reference Station* (VRS) e convertê-las em tempo útil para o formato RTCM SC-104;

- Implementar o conceito de IDGNSS, aplicando num nó central as correcções diferenciais no formato RTCM SC-104 às medidas brutas efectuadas pelos receptores e obtendo as respectivas posições corrigidas;
- Armazenar numa base de dados os dados de cada receptor;
- Disponibilizar a informação obtida numa interface do tipo aplicação *Web*, permitindo, assim, o seguimento dos receptores de qualquer ponto.

1.4 Estrutura do Relatório

Este relatório encontra-se estruturado em sete capítulos.

No Capítulo 1 de “Introdução” fornece-se ao leitor o contexto desta Dissertação, os objectivos estabelecidos para a sua prossecução e a motivação que levou à sua realização.

No Capítulo 2 do “Estado da Arte” apresentam-se os princípios de funcionamento da negação suportada por satélites e descrevem-se os principais sistemas de navegação e posicionamento bem como os seus sistemas de complementares.

No Capítulo 3 de “Equipamento e Protocolos” descreve-se o equipamento e apresentam-se os protocolos usados no desenvolvimento deste trabalho.

No Capítulo 4 de “Configuração do Ambiente de Desenvolvimento” apresenta-se uma descrição do processo de instalação e configuração do ambiente de desenvolvimento escolhido bem como as bibliotecas e aplicações reutilizadas neste trabalho.

No Capítulo 5 de “Desenvolvimento do Sistema” efectua-se uma descrição do desenvolvimento do sistema, evidenciando a estrutura do trabalho realizado e os passos realizados.

No Capítulo 6 de “Resultados dos Testes e Depuração do Código” caracteriza-se o funcionamento da aplicação desenvolvida, comparando resultados e verificando a inexistência de erros.

No Capítulo 7 de “Conclusão e Desenvolvimentos Futuros” analisa-se o trabalho desenvolvido face aos objectivos iniciais, descrevendo as principais conclusões obtidas e perspectivando-se alguns desenvolvimentos futuros.

Capítulo 2

Estado da Arte

Neste capítulo apresenta-se o estado da arte no domínio do posicionamento, nomeadamente no que concerne aos sistemas de navegação e posicionamento, sistemas de melhoramento da exactidão e sistemas diferenciais. Também será referido neste contexto o processo de transmissão da informação proveniente de sistemas de navegação e posicionamento assim como a minimização das fontes de erro que se encontram normalmente associadas a esses sistemas.

2.1 Sistemas de Navegação e Posicionamento

Sistemas Globais de Navegação por Satélite é o termo genérico que se utiliza para descrever os sistemas de navegação que fornecem aos utilizadores uma solução de posicionamento tridimensional, obtida por alinhamento passivo através de sinais de rádio transmitidos por satélites em órbita.

Estes sistemas encontram-se divididos naqueles cuja operação é global, como é o caso do sistema norte-americano *NAVigation Satellite Timing and Ranging Global Positioning System* (NAVSTAR-GPS), do sistema russo *Globalnaya Navigational Sustenizavam Sistema* (GLONASS) ou do sistema europeu Galileo; e naqueles cuja operação é regional, como é o caso do sistema chinês Beidou/Compass ou do sistema japonês *Quasi-Zenith Satellite System* (QZSS). Apesar de alguns destes sistemas estarem apenas parcialmente implementados, há características que são transversais a todos eles, nomeadamente a arquitectura e o processo de posicionamento.

2.1.1 Arquitectura

A arquitectura dos GNSS é composta por três segmentos: o segmento espacial, o segmento de controlo e o segmento de utilização que são ilustrados na Figura (2.1).

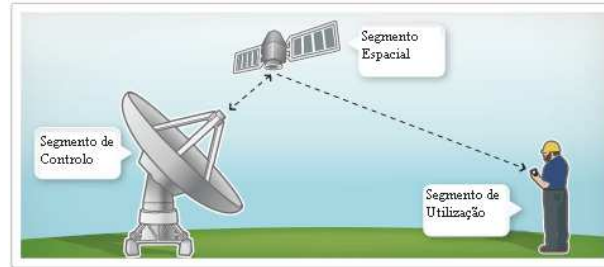


Figura 2.1: Arquitectura dos GNSS

O segmento espacial é o componente que difunde de forma constante e global a informação necessária ao funcionamento do sistema. Para tal, a constelação de satélites ou *space vehicles* (SV) que orbita em torno da Terra de cada sistema tem de garantir que, pelo menos, quatro satélites sejam simultaneamente visíveis de qualquer ponto da superfície terrestre. Os satélites emitem continuamente os sinais utilizados pelos receptores para a determinação da posição. Estes sinais são constituídos pela mensagem de navegação e por códigos de aquisição. Os códigos de aquisição permitem que os receptores sejam capazes de determinar o instante de tempo em que os sinais recebidos foram transmitidos. A mensagem de navegação inclui parâmetros temporais e informação relativa à constelação de satélites.

O segmento de controlo consiste numa rede de estações de monitorização, estações de controlo e antenas, responsáveis por controlar e monitorar o segmento espacial. As estações de monitorização, dotadas de relógios sincronizados, obtêm os sinais de posicionamento e utilizam-nos para determinar as órbitas e calibrar os relógios dos SV. As estações de controlo calculam a mensagem de navegação de cada satélite, determinam se é necessário realizar quaisquer ajustes à constelação e transmitem seguidamente esta informação para o segmento espacial através das antenas.

O segmento de utilização consiste nos receptores GNSS que são compostos pela antena, responsável por converter os sinais provenientes do segmento espacial em sinais eléctricos; pelo receptor, que desmodula os sinais através de uma referência de relógio; pelo processador de distâncias, que utiliza algoritmos de seguimento e aquisição para determinar a distância entre a antena e cada um dos satélites da constelação em uso; e pelo processador de navegação, que usa as

medidas determinadas pelo processador de distâncias para determinar a posição, velocidade e tempo, conhecidas como solução PVT.

2.1.2 Fluxo da Informação

A transmissão da informação entre as diversas entidades que constituem um sistema de navegação e posicionamento flui da seguinte forma:

- O segmento de controlo, através da sua monitorização constante da constelação de satélites, calcula a mensagem de navegação para cada um deles, verifica se são necessários ajustes à sua configuração e envia esta informação para o segmento espacial;
- Recebida a informação, o segmento espacial disponibiliza-a ao segmento de utilização (os receptores GNSS) sob a forma de códigos de aquisição e mensagens de navegação;
- Os utilizadores equipados com os receptores GNSS, em posse destes dados, podem então determinar o instante em eles foram transmitidos através dos códigos de aquisição e, com base nessa informação e nos parâmetros temporais e orbitais contidos na mensagem de navegação, obter a solução PVT.

2.1.3 Posicionamento

A solução de posicionamento dos GNSS é determinada por uma aquisição a três dimensões, em que o tempo de chegada do sinal t_{sa} é determinado pelo relógio do receptor e o tempo de transmissão de cada sinal t_{st} é obtido através do código de aquisição e pela mensagem de navegação [1].

Se os relógios do receptor e do satélite estivessem sincronizados e não fossem consideradas as restantes fontes de erro, a distância de um satélite ao utilizador ρ medida pelo segmento de utilização seria obtida pela diferença dos tempos de chegada e de transmissão, multiplicados pela velocidade da luz no vácuo - ver Equação (2.1).

Equação 2.1: Distância teórica entre satélite e receptor

$$\rho_j = (t_{sa,i} - t_{st,i}) * c \quad (2.1)$$

onde o índice i é utilizado para identificar o número do satélite em questão.

Se fosse medida apenas a distância entre o receptor e um único satélite, a posição do utilizador poderia ser uma qualquer sobre a superfície esférica de

raio ρ centrada nesse satélite. Se fossem utilizados dois satélites, a posição do utilizador resultaria da intersecção das duas superfícies esféricas de raios ρ_1 e ρ_2 , respectivamente, *i.e.*, sobre uma circunferência.

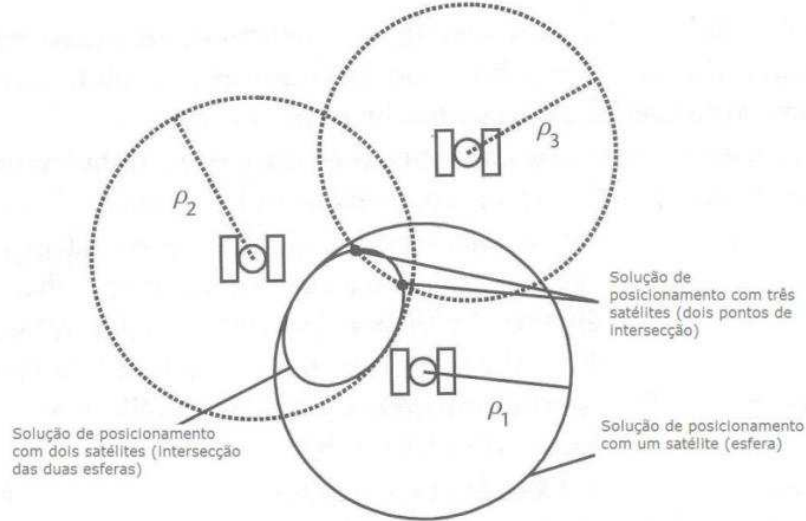


Figura 2.2: Posicionamento a uma, duas e três dimensões [1]

Ao adicionar as medidas a um terceiro satélite, a posição do utilizador ficaria limitada a dois pontos da circunferência, como é ilustrado na Figura (2.2). Para a maioria das aplicações é necessária apenas uma solução de posicionamento viável, ambiguidade essa que apenas é resolvida com a utilização de medidas a quatro satélites.

No entanto, na prática, os relógios do receptor e do satélite não se encontram sincronizados. Se o relógio do receptor estiver adiantado face ao relógio do satélite, o tempo de chegada medido $t'_{sa,j}$ surgirá atrasado face ao verdadeiro tempo de chegada, t_{sa} , o que resulta numa medida sobrestimada. Também, se o relógio do satélite estiver adiantado face ao relógio do receptor, o verdadeiro tempo de transmissão t_{st} resultará adiantado face ao tempo de transmissão pretendido $t'_{st,j}$, que é o deduzido pelo segmento de utilização através do código de aquisição e da mensagem de navegação. Deste modo, se o relógio do receptor estiver adiantado δt_{rc} e o relógio de um satélite j também estiver adiantado δt_{sj} , o erro de medida da distância não considerando outras fontes de erro é dada pela Equação (2.2).

Equação 2.2: Erro de medida da distância entre satélite e receptor

$$\delta\rho_j = \rho'_{Rj} - \rho_j = (\delta t_{rc} - \delta t_{sj}) * c \quad (2.2)$$

onde ρ'_{Rj} , que é a distância medida na presença de erros de relógio, é designada pseudodistância para se diferenciar da distância verdadeira ρ_j , que seria medida na ausência de erros de relógio.

Os erros de relógio dos satélites são medidos pelo segmento de controlo e transmitidos na mensagem de navegação; sendo assim o processador de navegação tem a capacidade de os corrigir. Já os erros de relógio dos receptores são completamente desconhecidos. No entanto, este parâmetro pode ser utilizado como uma incógnita na solução de posicionamento. Cada pseudodistância medida ρ'_{Cj} , já corrigida dos erros de relógio do satélite, pode ser expressa através da Equação (2.3).

Equação 2.3: Pseudodistância corrigida

$$\rho'_{Cj} = \sqrt{(r_{isj}^i(t_{st,j}) - r_{ia}^i(t_{sa}))^T (r_{isj}^i(t_{st,j}) - r_{ia}^i(t_{sa}))} + \delta\rho_{rc}(t_{sa}) \quad (2.3)$$

em que $r_{isj}^i(t_{st,j})$ é a posição do satélite j , $r_{ia}^i(t_{sa})$ é a posição da antena do receptor a e $\delta\rho_{rc} = \delta t_{rc} * c$ é o erro do relógio do receptor.

A posição do satélite é obtida por um conjunto de parâmetros difundidos na mensagem de navegação designado efemérides, que descrevem a órbita do satélite, em conjunto com a medida corrigida do tempo de transmissão de cada sinal. Sabendo que o vector de posição r pode ser decomposto nas suas componentes $r^i = (x^i, y^i, z^i)$, obtém-se a Equação (2.4).

Equação 2.4: Posição do receptor

$$\begin{aligned} \rho'_{Cj} &= \sqrt{\Delta x_j^2 + \Delta y_j^2 + \Delta z_j^2} + \delta\rho_{rc}(t_{sa}) \\ \Delta x_j &= x_{isj}^i(t_{st,j}) - x_{ia}^i(t_{sa}) \\ \Delta y_j &= y_{isj}^i(t_{st,j}) - y_{ia}^i(t_{sa}) \\ \Delta z_j &= z_{isj}^i(t_{st,j}) - z_{ia}^i(t_{sa}) \end{aligned} \quad (2.4)$$

que permite obter a solução de posicionamento a três dimensões. As quatro incógnitas são comuns às equações das pseudodistâncias a cada satélite, assumindo um tempo de chegada do sinal comum. Sendo assim, é possível resolver a solução de posicionamento com quatro equações, utilizando, como mostra a Figura (2.3), medidas relativas a quatro satélites. Também e da mesma forma, é possível obter a velocidade da antena do utilizador através de um conjunto de gamas de pseudodistâncias e gamas de variação de pseudodistâncias [1]. Em suma, a partir deste conjunto de parâmetros é possível obter a solução PVT e proporcionar soluções de posicionamento para diversas aplicações.

A solução de navegação GNSS é susceptível a uma série de fontes de erro, incluindo diferenças entre as efemérides difundidas e as verdadeiras efemérides,

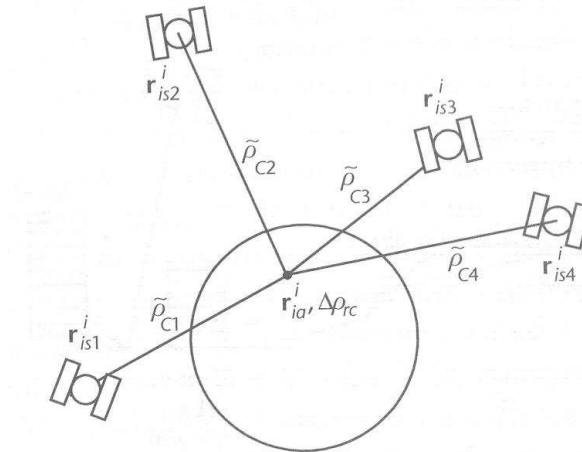


Figura 2.3: Solução de posicionamento com quatro satélites [1]

erros de relógio dos satélites, atrasos de propagação dos sinais devido às camadas atmosféricas e erros de medida no receptor. Assim sendo, estas soluções têm uma exactidão de apenas alguns metros. Esta exactidão pode ser melhorada com o auxílio de sistemas de calibração da informação, provenientes de uma ou mais estações de referência situadas em localizações conhecidas. Este sistema complementar é conhecido como GNSS diferencial ou *Diferencial Global Satellite Navigation System* (DGNSS). Este sistema é fundamental para um variado conjunto de aplicações, onde a exactidão do posicionamento é crucial e, por esse motivo, será abordado com mais detalhe na secção (2.3).

2.1.4 Fontes de Erro

Um receptor GNSS é capaz de obter a sua posição baseando-se no tempo decorrido entre a transmissão e a recepção dos sinais emitidos pelos satélites. No entanto, este processo está sujeito a um conjunto de desvios ou erros que interferem com a exactidão do sistema. Estes erros são, fundamentalmente, de três tipos: erros externos causados pelo meio de propagação, erros internos associados ao funcionamento do sistema e erros intencionais [3]. Os sinais, ao percorrerem a atmosfera, são absorvidos e atrasados pelas camadas ionosféricas e troposféricas e, ao atingirem a superfície terrestre, são facilmente reflectidos e refractados dando assim origem a fenómenos de multitrajeto. Também o sistema propriamente dito é responsável por erros e desvios nas medições efectuadas tais como a falta de exactidão das efemérides transmitidas, os desvios dos relógios dos receptores e dos satélites e o ruído electrónico dos receptores. Há ainda os erros intencionalmente provocados pelas entidades responsáveis pelos GNSS e os erros

aleatórios que não são previsíveis nem modeláveis e que também contribuem para os desvios na exactidão. Assim, podem-se identificar as seguintes fontes de erro:

- Erros nos relógios dos satélites - Os relógios que se encontram nos satélites são relógios atómicos de elevada precisão. No entanto, não são perfeitos e como tal provocam erros na solução de navegação calculada pelos receptores. Para minimizar o efeito destes erros, o seu funcionamento deve ser continuamente monitorado pelas estações de controlo, de modo a prevenir desvios e incluir os devidos parâmetros de correcção na mensagem de navegação a enviar aos receptores.
- Erros no relógio do receptor GNSS - Os relógios dos receptores GNSS são, de um modo geral, pouco precisos, provocando assim erros consideráveis e bastante superiores aos provocados pelos relógios dos satélites. No entanto, como este erro é comum a todas as observações, pode ser quantificado e eliminado recorrendo a observações de quatro satélites para a determinação da posição do receptor.
- Erros nas efemérides - As órbitas descritas pelos satélites são consideradas regulares, pelo que podem ser descritas por um modelo matemático rigoroso. No entanto, devido aos efeitos gravitacionais do Sol e da Lua, existem variações nas órbitas dos satélites impossíveis de medir e compensar. Este efeito pode ser minimizado através de uma monitorização constante por parte das estações terrestres, mas dada a impossibilidade de eliminar eventuais erros, estes acabam por ser transmitidos na mensagem de navegação, não havendo forma de os compensar.
- Atrasos troposféricos - Os sinais provenientes dos satélites sofrem refacções ao atravessar a troposfera, provando assim atrasos que podem chegar até aos 3 m. Este atraso é mais significativo em satélites que se encontram perto da linha de horizonte do receptor, uma vez que os seus sinais atravessam durante mais tempo a troposfera [3]. Este atraso pode ser modelado através da medição das componentes húmida e seca presentes na troposfera e minimizado através da selecção de satélites com um ângulo de elevação superior a 15° (normalmente, opta-se por escolher uma máscara de 10° , no sentido de não bloquear demasiados satélites).
- Atrasos ionosféricos - O atraso provocado pela refacção dos sinais durante o atravessamento da ionosfera depende da actividade solar, do período do dia e da geometria dos satélites. O erro da posição resultante causado por este factor pode atingir os 30 m durante o dia, os 6 m durante a noite, é superior para os sinais de satélites situados perto do horizonte e inferior para os localizados sobre o zénite do receptor [3]. Estes erros, apesar de não serem previsíveis, podem ser estimados através de modelos ionosféricos.

- Erros provocados por reflexão, refração e multitrajeto - O trajeto que os sinais fazem desde o satélite até ao receptor GNSS nem sempre é em linha recta. A existência de obstáculos pode provocar reflexões ou refrações no sinal, dando-se assim o fenómeno de multitrajeto e consequente interferência no sinal que chega ao receptor. A colocação do receptor num local elevado em relação ao ambiente circundante ou a aplicação de máscaras de selecção de satélites com ângulo de elevação superior a 15° bem como a recolha de várias observações no mesmo local durante um certo intervalo de tempo são formas de minimizar estes erros.
- Ruído do receptor GNSS - O receptor GNSS não é um equipamento ideal e apresenta várias limitações, nomeadamente ruído electrónico, resolução do equipamento, *etc.*
- Mecanismos de degradação intencional - Mecanismos dos GNSS militares destinados a degradar intencionalmente a qualidade de serviço de terceiros.

2.1.5 Sistemas de Navegação e Posicionamento Global

Os sistemas globais de navegação por satélite encontram-se, como já foi referido, divididos naqueles cuja cobertura é global e cuja cobertura é regional. Seguidamente serão descritas as características dos sistemas de cobertura global que se encontram totalmente implementados como é o caso do sistema norte-americano NAVSTAR-GPS e do sistema russo GLONASS e também dos que se encontram em fase de implementação, como é o caso do sistema europeu Galileo.

2.1.5.1 NAVigation Satellite Timing and Ranging Global Positioning System

O sistema NAVSTAR-GPS, cujo logótipo se encontra ilustrado na Figura (2.4), é vulgarmente conhecido por GPS. Foi desenvolvido pelo governo dos Estados Unidos da América (EUA) para fins militares e é controlado pelo GPS Wing, antigamente conhecido como *Joint Program Office* (JPO), sob o comando do Departamento de Defesa ou *Department of Defense* (DoD) [14].

2.1.5.1.1 Serviços

O GPS oferece dois serviços de navegação: o serviço convencional para utilização civil conhecido como *Standard Positioning Service* (SPS) e o serviço preciso para utilização militar designado *Precise Positioning Service* (PPS). O SPS encontra-se disponível para todos os utilizadores que possuam equipamento adequado; já o PPS utiliza sinais encriptados e apenas se encontra disponível a utilizadores licenciados pelo governo norte-americano. Durante os anos 90, a



Figura 2.4: Logótipo do sistema NAVSTAR-GPS [2]

exactidão do SPS foi propositadamente degradada em cerca de 100 m, através de uma técnica designada *Selective Availability* (SA) para impedir o posicionamento exacto de forças consideradas hostis. No entanto, o mecanismo SA pode ser ultrapassado através da utilização de posicionamento diferencial, já referido no ponto anterior, tendo sido por esta razão suspenso em 2000. Assim e actualmente, o SPS básico oferece uma exactidão horizontal de cerca de 3,8 m e vertical de cerca de 6,2 m, enquanto o PPS oferece uma exactidão horizontal de cerca de 1,2 m e vertical de cerca de 1,9 m [1].

2.1.5.1.2 Segmento Espacial

O GPS opera actualmente com uma constelação composta por 24 satélites operacionais que orbitam a Terra num raio de 26 600 km (20 100 km acima da superfície terrestre) com um período de 11 h e 58 min [1]. A constelação está organizada em seis planos orbitais, como pode se ver pela Figura (2.5), cada um com pelo menos quatro satélites, o que perfaz a totalidade dos 24 satélites, garantindo que são sempre visíveis de qualquer ponto da superfície terrestre entre 5 a 14 satélites (assumindo situações em linha de vista). O segmento de controlo compreende, por seu lado, uma estação central designada por *Master Control Station* (MCS), situada no Colorado em Colorado Springs, seis estações de monitorização e quatro estações para envio de dados. O sistema pode suportar um máximo de 36 satélites operacionais.

2.1.5.1.3 Sinais

O GPS difunde 10 sinais de navegação em três faixas de frequência conhecidas por L1 (com frequência 1575,42 MHz), L2 (com frequência 1227,60 MHz) e L5 (com frequência 1176,45 MHz), recorrendo à técnica *Code Division Multiple Access* (CDMA). Os satélites da constelação GPS podem também difundir sinais a 1381,05 MHz (L3) que não se destinam à navegação [1]. Os dois sinais mais significativos deste sistema são o código *coarse/acquisition* (C/A), utilizado pelo

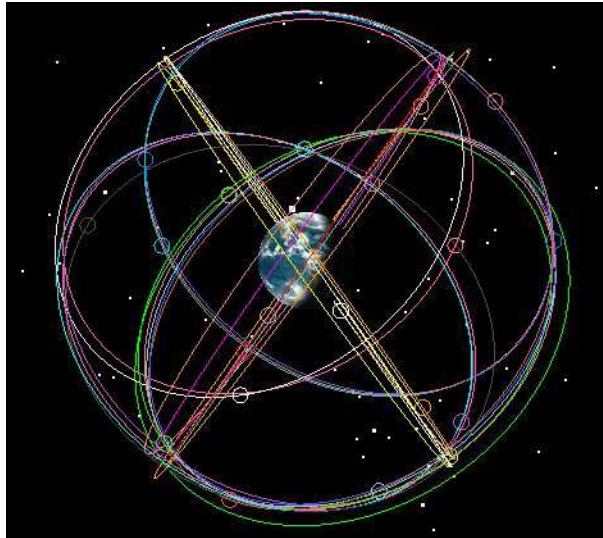


Figura 2.5: Órbitas do sistema NAVSTAR-GPS

SPS e PPS e difundido na faixa L1, e o código *encrypted precise* (P(Y)), apenas utilizado pelo PPS e difundido nas faixas L1 e L2. O código C/A foi concebido para fornecer soluções de navegação de menor exactidão, daí estar disponível para todos os utilizadores do SPS, ao contrário do código P(Y), que é caracterizado por sinais precisos multiplicados por uma chave de encriptação apenas disponível para utilizadores PPS licenciados. De notar que, actualmente, o SPS conta também com o sinal L2C, permitindo assim a difusão de sinais para utilização civil a duas frequências, tornando-os mais robustos e menos sensíveis às perturbações das camadas atmosféricas.

Os sinais C/A e P(Y), apesar de possuírem características distintas, são códigos do tipo pseudo-aleatório gerados por computadores a bordo dos satélites. O sinal C/A, também designado por código de aquisição C/A ou *Pseudorandom Noise* (PRN), é transmitido a uma taxa de 1,023 Mb/s e tem 1023 b de comprimento. Cada satélite emite o seu código PRN único, o que permite a sua inequívoca identificação. O sinal P(Y), que é transmitido a uma taxa de 10,23 Mb/s, é um código muito longo, com a duração de 266 d, estando dividido em segmentos de 7 d. Cada satélite apresenta um segmento semanal único, que muda todas as semanas e que o distingue dos restantes. Este código designa-se por P(Y) porque, antes de ser emitido, é intervalado com um outro código, o código Y, que actua como mecanismo anti-fraude [3][15].

2.1.5.1.4 Mensagem de Navegação

Os satélites GPS difundem três diferentes mensagens de navegação, dependendo

dos sinais a serem utilizados: a mensagem de navegação convencional, difundida simultaneamente nos sinais C/A e P(Y); a mensagem de navegação militar, difundida nos sinais *military* (M); e a mensagem de navegação comercial, difundida nos sinais *civil moderate* (CM). Enquanto as efemérides e os dados de calibração dos relógios dos satélites são difundidos cada 30 s, o almanaque (que compreende parâmetros aproximados das efemérides, calibração de relógios e do estado de toda a constelação) apenas é difundido cada 12,5 min. Como se pode ver, os dados do almanaque são válidos por um período de tempo muito maior que as efemérides, pois o almanaque apenas serve como auxílio ao segmento de utilização no que toca à escolha dos satélites para aquisição dos sinais.

A mensagem de navegação contém informação rigorosa relativa às correcções a aplicar ao relógio e às efemérides dos satélites bem como dados relativos à constelação, nomeadamente, o almanaque e o estado de saúde da globalidade dos satélites. Esta informação é de grande precisão e referente a cada satélite do sistema, possibilitando assim a obtenção da posição de um receptor GNSS. Cada mensagem de navegação é composta por 25 tramas, sendo cada trama de 1,5 kb de comprimento e é constituída por 5 subtramas de 300 b. As subtramas são, por sua vez, constituídas por 10 palavras de 30 b e cada palavra é composta por 24 b de dados e 6 b de paridade. É gerada a uma taxa de 50 b/s e tem uma duração de 12,5 min. A estrutura genérica destas mensagens encontra-se ilustrada na Figura (2.6).

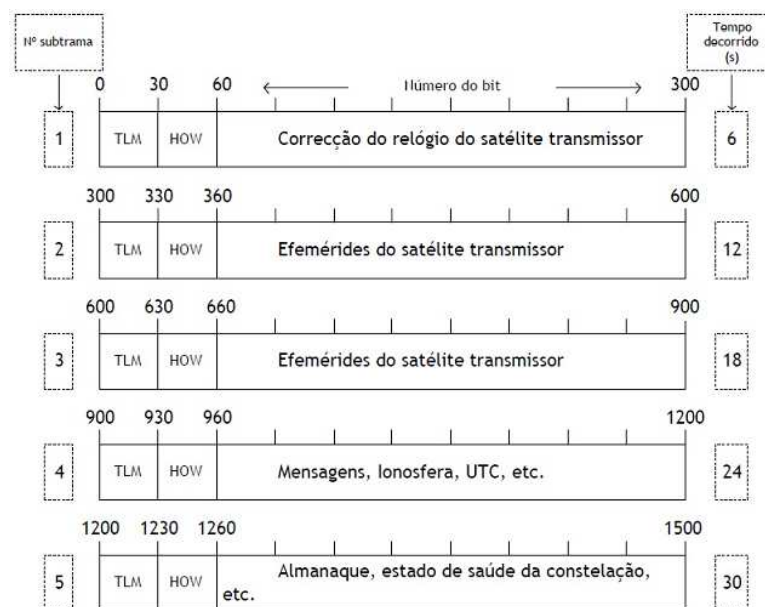


Figura 2.6: Estrutura de uma mensagem de navegação [3]

A primeira e a segunda palavra de cada subtrama contêm, respectivamente,

os dados de telemetria do satélite, *Telemetry Word* (TLM), e de *handover*, *Handover Word* (HOW), que permite aos utilizadores do serviço PPS transitarem do serviço de posicionamento padrão (que utiliza código C/A) para o serviço de posicionamento preciso (que utiliza código P(Y)). As restantes palavras têm dados que variam conforme a subtrama e a trama a que pertencem. Enquanto as três primeiras subtramas de cada trama contêm os dados relativos ao satélite emissor, as duas últimas subtramas de cada trama comportam os dados sobre a constelação. Os dados relativos ao satélite são essenciais no que diz respeito à determinação da posição de um receptor GNSS. Estes dados incluem os parâmetros de correcção dos desvios dos relógio dos satélites, do estado de saúde dos satélites e ainda dados rigorosos relativos às efemérides dos satélites. Estes dados são repetidos dentro de cada mensagem a cada 30 s. A informação sobre a constelação inclui o almanaque, que informa o receptor acerca da localização e dos parâmetros sobre o estado de todos os satélites da constelação (nomeadamente configuração e saúde), referências temporais e parâmetros relativos aos modelos das camadas atmosféricas, mais precisamente a ionosfera, permitindo que o receptor GNSS estime o atraso provocado pelo atravessamento dessa camada. Assim, a constituição das tramas de uma mensagem de navegação pode ser definida do seguinte modo:

- Subtrama 1 transporta os parâmetros de correcção dos desvios dos relógio dos satélites bem como informação sobre a referência semanal (*GPS Week*) e o estado de saúde do satélite;
- Subtramas 2 e 3 transportam dados rigorosos relativos às efemérides dos satélites;
- Subtramas 4 e 5 transportam as páginas do almanaque da constelação, assim como referências temporais e os parâmetros do modelo ionosférico.

De notar que os parâmetros das efemérides mantêm-se válidos por apenas 2 h, ao passo que o almanaque tem uma validade de alguns meses. É por este motivo que o almanaque não é adequado para a determinação da solução PVT de um receptor GNSS, permitindo apenas prever que satélites poderão estar disponíveis num dado local e num dado tempo específico. Para a determinação de uma solução de posicionamento, como já foi referido, é necessária informação precisa e rigorosa, o que só é conseguido através das efemérides dos satélites.

2.1.5.1.5 Tipos de Observações

Os receptores efectuam dois tipos de observações: o seguimento da fase do código modulador (C/A ou P) e o seguimento da fase da portadora. No caso de aplicações civis, enquanto os receptores de frequência única (L1) apenas utilizam

as observações do código C/A para determinar a solução PVT, os receptores de dupla frequência (L1 e L2) utilizam preferencialmente as observações de fase da portadora. O princípio teórico de funcionamento dos dois tipos de observação é o mesmo: o receptor determina o tempo de transmissão (o tempo decorrido entre a emissão e a recepção) do sinal respectivo (código modulador ou portadora) e multiplica-o pela velocidade da luz no vácuo. Como o comprimento de onda do código C/A é 29,326 m e o da portadora L1 é 19,04 cm, a exactidão da solução PVT resultante no segundo caso será sempre muito superior. No entanto, neste último caso, existe um parâmetro desconhecido designado factor de ambiguidade, que representa o número de ciclos completos que a portadora realizou durante o trajecto do satélite ao receptor, que apenas os receptores de dupla frequência têm a capacidade de determinar.

Os receptores GNSS determinam o tempo de transmissão através da replicação local das portadoras e dos códigos pseudo-aleatórios emitidos pelos satélites. No caso específico do seguimento de um código pseudo-aleatório, o receptor, primeiro, descodifica a mensagem de navegação e gera uma réplica do código sincronizada com o satélite e, de seguida, determina a descorrelação temporal entre a réplica gerada e o sinal recebido, *i.e.*, o tempo de transmissão do sinal. Este processo designa-se por “Aquisição Inicial de Posição”, na medida em que o receptor GNSS obtém a sua localização baseando-se no tempo decorrido entre a emissão e a recepção do sinal.

2.1.5.1.6 Mecanismos de Degradação Intencional

O NAVSTAR-GPS, na qualidade de GNSS militar, possui um conjunto de mecanismos de degradação intencional do serviço. O mecanismo de disponibilidade selectiva, anteriormente referido como SA, consiste na adição de desvios intencionais aos dados emitidos dos satélites, resultando num erro superior a 100 m. Apesar de actualmente estar desactivado, pode a qualquer momento ser reactivado pelo governo dos EUA. O mecanismo anti-fraude ou *anti-spoofing* (A-S) impede os que os utilizadores convencionais do NAVSTAR-GPS acedam ao código preciso e, conseqüentemente, obtenham maior exactidão.

Desde que foi declarado operacional em 1993, o NAVSTAR-GPS foi alvo de uma série de melhoramentos em todos os seus segmentos: espacial, de controlo e de utilização [16]. No segmento espacial tem havido melhoramentos em termos de potência dos sinais, exactidão dos relógios dos satélites e fiabilidade. No segmento de controlo tem havido melhoramentos na modelação de órbitas e das camadas atmosféricas, diminuindo assim os erros e permitindo um melhoramento significativo na exactidão do posicionamento final. No segmento de utilização, quer no sector civil, quer no sector militar, tem havido sucessos extraordinários. Os fabricantes de equipamentos para utilização civil têm colocado no mercado

produtos portáteis, pequenos e baratos (como por exemplo, telemóveis com GPS integrado) e as pessoas têm sido receptivas a estes equipamentos. Também no campo militar, quase todo o equipamento bélico vem equipado com receptores GNSS para aumentar a sua precisão.

2.1.5.2 Globalnaya Navigational Sustenizavam Sistema

O sistema GLONASS, cujo logótipo se encontra ilustrado na Figura (2.7), foi desenvolvido também para fins militares pela antiga União das Repúblicas Socialistas Soviéticas (URSS) nos anos 70, paralelamente ao sistema norte-americano NAVSTAR-GPS.



Figura 2.7: Logótipo do sistema GLONASS [4]

Tal como o NAVSTAR-GPS, o GLONASS foi também concebido para oferecer serviços de posicionamento para fins civis e militares. Com a dissolução da URSS, este sistema foi continuado pela Rússia e atingiu a totalidade da sua constelação em 1995. No entanto e devido a problemas financeiros, a constelação veio a diminuir com o passar dos anos, até que, em 2001, o programa foi revitalizado e, com a colaboração da Índia, prevê-se que venha a estar totalmente operacional em 2010 [1].

2.1.5.2.1 Segmento Espacial

O segmento espacial GLONASS é constituído por 24 satélites (21 estão operacionais e 3 são sobresselentes) que orbitam a Terra num raio de 25 600 km (19 100 km acima da superfície terrestre) e se encontram distribuídos em três planos orbitais com oito satélites cada como mostra a Figura (2.8). O segmento de controlo compreende uma estação central nos arredores de Moscovo, duas estações de monitorização também em Moscovo e quatro estações para telemetria, seguimento e controlo ou *telemetry, tracking and control* (TTC). As estações TTC são responsáveis pelo *uplink* de comandos para os satélites e medida das órbitas

por radar, enquanto as estações de monitorização são apenas utilizadas para calibrar os relógios dos satélites. Actualmente estão previstos melhoramentos no GLONASS, nomeadamente, o aumento do número de estações de monitorização, o que fará com que este sistema se iguale ao GPS em termos de exactidão (o que se espera ser atingido no início de 2011) [1].

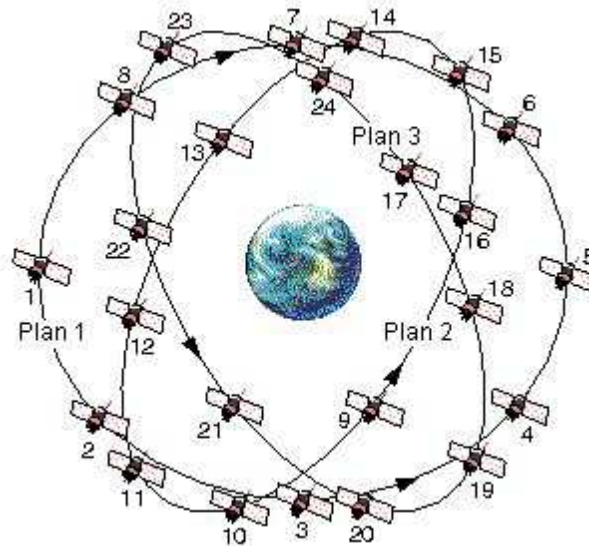


Figura 2.8: Órbitas do sistema GLONASS [5]

2.1.5.2.2 Sinais

O sistema GLONASS utiliza a técnica *Frequency Division Multiple Access* (FDMA) e faz uso de faixas L1 e L2, um pouco acima das congéneres L1 e L2 do GPS, e da faixa L3 que se situa entre a L2 e L5 do GPS [1]. A cada satélite é alocado um canal k que difunde, se for na faixa L1, a $(1602 + 0,5625k)$ MHz e, se for na L2, a $(1246 + 0,4375k)$ MHz.

2.1.5.2.3 Mensagem de Navegação

Em termos da mensagem de navegação, o GLONASS difunde diferentes mensagens de navegação nos sinais C/A e P. As efemérides e a informação de relógio dos satélites é difundida a cada 30 s no código C/A e a cada 10 s no código P. O almanaque é repetido a cada 2,5 min no código C/A e a cada 12 min no código P.

Este sistema, tal como já foi referido, tem sido alvo de uma reestruturação e, de acordo com o governo russo, inclui não só o segmento espacial, mas também o segmento de controlo. Em termos dos SV, espera-se que a constelação cresça e

que os SV tenham um tempo de vida entre 10 a 12 anos [16]. Adicionalmente, a Rússia pretende tornar o GLONASS interoperável com o GPS e o futuro Galileo, fornecendo mecanismos que permitam à sua indústria fabricar equipamentos receptores capazes de escutar os diferentes sistemas GNSS. Em termos do segmento de controlo, o acordo de cooperação realizado com a Índia vai permitir desenvolver e alargar as estações de TTC bem como facilitar as operações de manutenção da constelação através do uso de veículos de lançamento espacial indianos.

2.1.5.3 Galileo

O desenvolvimento do sistema de navegação Galileo da União Europeia, cujo logótipo se encontra ilustrado a Figura (2.9), teve início em 1999 e está a cargo da *European Space Agency* (ESA). O seu primeiro satélite, o GIOVE-A, foi lançado em Dezembro de 2005 e o segundo, o GIOVE-B, em Abril de 2008. Espera-se que os primeiros quatro satélites sejam colocados em órbita entre 2010 e 2012 e que a constelação esteja completamente operacional em 2014.



Figura 2.9: Logótipo do sistema Galileo [6]

2.1.5.3.1 Serviços

Ao contrário dos sistemas NAVSTAR-GPS e GLONASS, o Galileo foi desenvolvido unicamente para utilização civil, oferecendo quatro diferentes tipos de serviços de navegação: o serviço aberto ou *open service* (OS); o serviço *safety-of-life* (SOL); o serviço público regulado ou *public regulated service* (PRS); o serviço comercial ou *commercial service* (CS); e o serviço de busca e salvamento ou *search-and-rescue* (SAR) [1].

O OS oferece sinais em duas faixas de frequência para todos os utilizadores, sem qualquer custo adicional, com exactidão horizontal na ordem dos 2 m e vertical na ordem dos 4 m. O serviço SOL utiliza os mesmos sinais do OS, mas adiciona mecanismos de integridade e autenticação dos dados que asseguram a autenticidade do sinal Galileo (protegendo-o de *spoofing*). O CS fornece exactidão

acrescida para os que estejam dispostos a pagar uma taxa de subscrição, oferecendo sinais encriptados com informação de elevada exactidão e garantindo assim qualidade de serviço. Quanto ao PRS, este é destinado a serviços de emergência e segurança dos estados membros da União Europeia, oferecendo sinais com elevada integridade e resistência a interferências, mas degradando um pouco a exactidão que, neste serviço, ronda os 3 m na horizontal e 6 m na vertical. O SAR permitirá, nomeadamente, receber pedidos de socorro emitidos por radiobalizas de emergência localizadas em navios, aeronaves ou pessoas e encaminhá-los para os centros regionais de busca e salvamento.

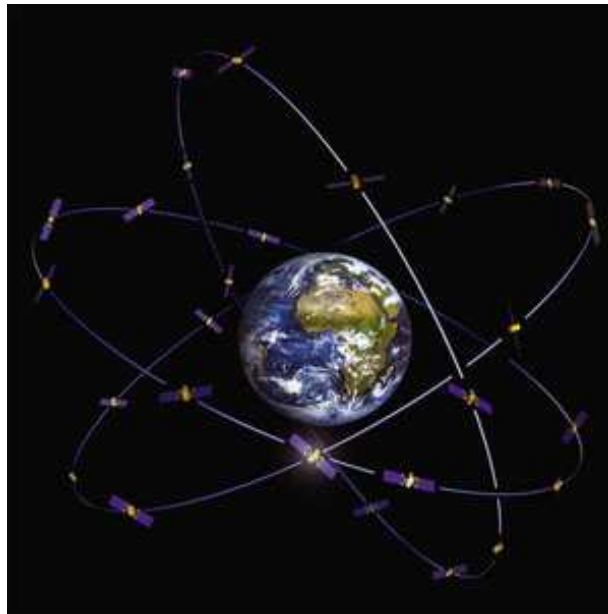


Figura 2.10: Órbitas do sistema Galileo [6]

2.1.5.3.2 Segmento Espacial

O Galileo será constituído por uma constelação de 27 satélites situados a 29 600 km (23 200 km acima da superfície terrestre) e distribuídos em três planos orbitais como mostra a Figura (2.10). Em termos de segmento de controlo, este sistema está dividido em dois subsistemas distintos: o *ground control segment* (GCS), situado na Alemanha, que é responsável por controlar o *hardware* dos satélites e o *ground mission segment* (GMS), situado em Itália, que é responsável por controlar o serviço de navegação (o centro de reserva quer do GCS, quer do GMS será localizado em Espanha). Em termos de constituição, o GCS será constituído por uma rede de cinco estações TTC e o GMS por uma rede de 10 estações para *uplink*, algumas colocadas em estações TTC, e uma rede global de 30 *Galileo Sensor Stations* (GSS), utilizadas para monitorar os sinais Galileo.

2.1.5.3.3 Sinais

Relativamente a sinais, o Galileo difunde 10 sinais de navegação diferentes em três faixas de frequência através da técnica CDMA: E5 (centrada a 1191,795 MHz), E6 (centrada a 1176,45 MHz) e E1-L1-E2 (centrada a 1207,14 MHz). A faixa E5 encontra-se dividida em duas subfaixas, E5a e E5b, sendo que a E5a coincide com a faixa L5 do sistema GPS e a E5b coincide com a faixa L3 do GLONASS. Já a faixa E1-L1-E2 coincide com a faixa L1 do GPS. Os restantes sinais, E6C, E6P e L1P, são sinais encriptados e utilizados nos serviços PRS, CS e SOL [1].

2.1.5.3.4 Mensagem de Navegação

Quanto às mensagens de navegação, o Galileo difunde quatro mensagens distintas: a mensagem acessível gratuitamente (difundida no sinal E5a-d), a mensagem de integridade (difundida nos sinais E5b-d e L1F), a mensagem comercial (difundida no sinal E6C-d) e a mensagem de acesso governamental (difundida em todos os sinais do PRS) [1]. Os dados de navegação são difundidos em todos os tipos de mensagens, com excepção da mensagem comercial; os dados das efemérides e do almanaque são semelhantes aos da mensagem convencional do sistema GPS e os dados de integridade e autenticação são emitidos somente nas mensagens de integridade e acesso governamental.

Existem poucos sistemas tecnológicos a ser alvo de tanta publicidade e interesse como o Galileo. No entanto, os sucessivos atrasos na sua implementação estão a provocar um certo descrédito, levantando a questão de se alguma vez a constelação estará totalmente operacional. Também o facto deste sistema ser totalmente vocacionado para a utilização civil, sem custos, levanta a questão da sustentabilidade financeira de um sistema que terá pouco retorno, pois só certos serviços são cobrados. Actualmente, aguarda-se com expectativa a evolução do projecto nos próximos 5 anos, nomeadamente o lançamento do número mínimo de satélites necessários para colocar o sistema operacional.

2.1.6 Sistemas de Navegação e Posicionamento Regional

Como já foi referido, existem também implementações de sistemas de posicionamento e navegação regionais designadas *Regional Satellite Navigation Systems* (RNSS). Actualmente existem alguns países a apostar neste tipo de implementações, sendo as mais significativas o RNSS chinês Beidou/Compass e o japonês QZSS, cujas principais características serão descritas seguidamente.

2.1.6.1 Beidou/Compass

O sistema de navegação Beidou utiliza uma constelação de três satélites geostacionários localizados nas longitudes E 80,2°, E 110,4° e E 139,9° [1]. O seu primeiro satélite foi lançado em 2000 e o sistema tornou-se totalmente operacional em 2004, destinando-se fundamentalmente a aplicações rodoviárias, ferroviárias e marítimas.

Para evitar a necessidade de sincronizar os relógios dos receptores, este sistema utiliza medidas a duas dimensões em que o receptor grava um pequeno segmento do sinal de navegação e reenvia-o para o satélite como um atraso fixo. Este parâmetro é então enviado para o centro de controlo, que através de uma base de dados de alturas do terreno, obtém a latitude e longitude do receptor e remete-as via satélite para o utilizador em questão. Este processo encontra-se ilustrado na Figura (2.11).



Figura 2.11: Posicionamento no sistema Beidou [7]

Os satélites Beidou difundem sinais de navegação a 2491,75 MHz, enquanto os equipamentos receptores transmitem a 1615,68 MHz, atingindo exactidões de 20 m em áreas cobertas pela rede de estações de calibração do sistema.

Deste modo, o Beidou proporciona à China um sistema de navegação por satélite completamente independente e controlável. No entanto e devido a limitações em termos de número de utilizadores, a China iniciou um programa de desenvolvimento de um novo sistema de navegação, o Compass, que será um sistema global e fará uso dos satélites do sistema regional.

2.1.6.2 Quasi-Zenith Satellite System

O sistema de navegação regional QZSS do Japão encontra-se ainda em projecto e irá, numa primeira fase, fornecer a utilizadores japoneses, embora a cobertura dos sinais se estenda a toda a Ásia Oriental e Oceânia, serviços de posicionamento para aplicações móveis e pessoais. Este serviço de posicionamento destina-se a complementar os sinais GPS, aumentando o número de satélites visíveis e fornecendo correcções diferenciais [17].

A constelação QZSS será composta por três satélites, situados em órbitas geostacionárias inclinadas 45° relativamente ao Equador, tendo sido o primeiro satélite da constelação, o MICHIBIKI, lançado em Setembro de 2010 [17]. Estes satélites serão colocados de forma a manter uma área de cobertura com a forma de um oito sobre a região da Ásia-Pacífico como se pode ver na Figura (2.12), o que permitirá que seja sempre visível pelo menos um satélite no território japonês.



Figura 2.12: Órbita do sistema QZSS [8]

2.2 Sistemas de Melhoramento

Os sistemas de melhoramento ou sistemas complementares procuram complementar os GNSS proporcionando serviços de posicionamento com sinais adicionais,

correções diferenciais e alertas de integridade, cruciais para aplicações em que a exactidão obtida directamente dos GNSS não é suficiente. O posicionamento utilizando sinais adicionais melhora a cobertura em áreas urbanas e montanhosas onde os sinais podem sofrer interferência devido aos obstáculos; as correções diferenciais possibilitam aumentos significativos na exactidão das soluções de navegação; e os alertas de integridade protegem os utilizadores de sinais erróneos.

Existem dois tipos de sistemas de melhoramento que se caracterizam fundamentalmente pela área que cobrem. O sistema de melhoramento baseado em satélites ou *space-based augmentation system* (SBAS) é destinado a servir áreas alargadas através da difusão de sinais a partir de satélites geostacionários. O sistema baseado em estações terrestres ou *ground-based augmentation system* (GBAS) é destinado a servir áreas locais através da transmissão de sinais de estações terrenas. Estes dois tipos serão seguidamente descritos com mais pormenor.

2.2.1 Sistemas de Melhoramento Suportados por Satélites

Os SBAS utilizam uma rede terrestre de estações de monitorização para realizar medidas que são enviadas para as estações principais através de uma rede de área alargada. Uma vez recebidos os dados das estações de monitorização, as estações principais usam essas medidas para gerar parâmetros de correcção das órbitas, relógios e modelos das camadas atmosféricas. Os parâmetros, assim que são validados, são enviados para os satélites geostacionários que, por sua vez, os difundem para que os utilizadores corrijam as respectivas soluções de navegação. Este processo encontra-se ilustrado na Figura (2.13).

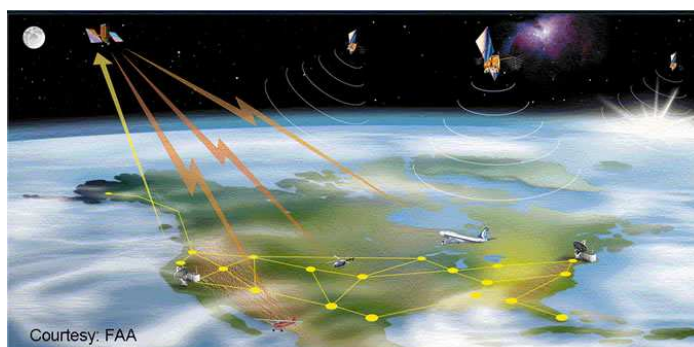


Figura 2.13: Arquitectura do SBAS [9]

Analisando o sistema, o SBAS apresenta três grandes evoluções em termos de desempenho. A primeira é o facto das correções diferenciais, como já foi referido,

aumentarem significativamente a exactidão das soluções de posicionamento. A segunda reside nos sinais adicionais que são tidos em conta no posicionamento, que não só aumentam a disponibilidade do sistema, mas também aumentam a exactidão. Por último, a validação dos sinais através da realização de testes de integridade melhora significativamente a qualidade do serviço prestado, alertando os utilizadores num prazo de 6 s caso haja alguma falha com o GNSS ou com o SBAS.

Em termos de implementação, há vários SBAS em funcionamento: o norte-americano *Wide-Area Augmentation System* (WAAS), o europeu *European Geostationary Navigation Overlay System* (EGNOS), o russo *System for Differential Correction and Monitoring* (SDCM), o japonês *Multi-Functional Satellite Augmentation System* (MSAS) e o indiano *GPS Aided Geo Augmented Navigation* (GAGAN). Cada um destes sistemas utiliza uma rede terrestre de estações de monitorização situada na sua área de cobertura. Isto implica que o serviço de melhoramento (com sinais adicionais, correcções diferenciais e verificação de integridade) apenas se encontra disponível na área coberta pela rede de estações de monitorização, independentemente do *footprint* ou área de cobertura do sistema. Assim e como se pode deduzir, basta adicionar estações de monitorização à rede para aumentar a cobertura do serviço prestado por cada SBAS. Também é de referir que todos os SBAS existentes transmitem na faixa L1 mensagens no mesmo formato, o formato *Radio Technical Commission for Aeronautics* (RTCA) DO-229.

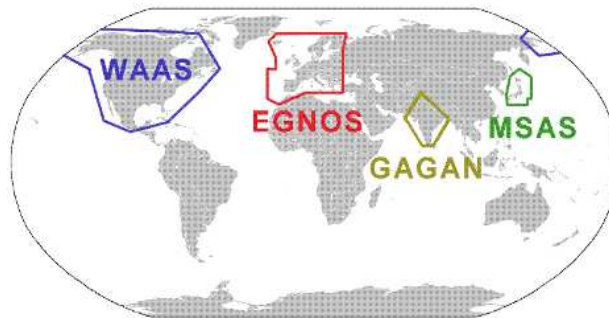


Figura 2.14: Cobertura dos diversos SBAS existentes [9]

Nos pontos seguintes serão descritos e caracterizados cada um dos sistemas de melhoramento baseado em satélites actualmente implementados. Apenas de notar que, e como se pode ver na Figura (2.14), estas implementações proporcionam actualmente um serviço de melhoramento com cobertura quase global.

2.2.1.1 Wide-Area Augmentation System

O sistema de melhoramento de área alargada WAAS foi desenvolvido em 2000 pela *Federal Aviation Administration* (FAA), entidade responsável pela aviação civil nos EUA, para responder aos requisitos de exactidão e integridade necessários à aeronavegação que não são satisfeitos pelo posicionamento em modo absoluto do NAVSTAR-GPS. Assim, o WAAS é um sistema que complementa o GPS, proporcionando-lhe maior exactidão, integridade e disponibilidade [3].

O WAAS é constituído por uma constelação de quatro satélites geostacionários e uma rede terrestre de 38 estações de monitorização ou referência, que cobrem uma grande área geográfica. Estas estações de referência de área alargada ou *wide-area reference stations* (WRS), encontram-se georreferenciadas com grande exactidão e equipadas com receptores NAVSTAR-GPS de elevada qualidade, sendo capazes de receber, calcular e difundir as correcções a aplicar à informação disponibilizada pelo GNSS norte-americano. Cada WRS da rede de monitorização transmite os seus dados para a estação principal de área alargada ou *wide-area master station* (WMS) que, por sua vez, processa a informação de correcção, aplica os algoritmos de correcção e garante a integridade do sistema. As mensagens de correcção são, então, geradas e enviadas para a constelação geostacionária do WAAS e, daí, difundidas na faixa L1 para os receptores que se encontram dentro da área de cobertura deste serviço de melhoramento.

2.2.1.2 European Geostationary Navigation Overlay System

O EGNOS, desenvolvido pela ESA em 2000, é um serviço europeu de melhoramento de área alargada, semelhante ao WAAS, que se destina a providenciar correcções de área alargada e integridade através de satélites geostacionários [3]. Este serviço complementar difunde e garante sinais de navegação para aplicações dinâmicas do tipo aeronáutico, marítimo e terrestre para os GNSS NAVSTAR-GPS e GLONASS.

O EGNOS é constituído por uma constelação de três satélites geostacionários, uma rede de área alargada de 34 estações de monitorização da integridade dos receptores ou *receiver integrity monitoring stations* (RIMS), cujas observações são posteriormente enviadas para o centro de controlo principal ou *master control center* (MCC). Este MCC é responsável por processar a informação e gerar as correcções, enviando-as para os satélites EGNOS, tal como a WMS do WAAS. A constelação é constituída pelo Inmarsat AOR-E (PRN 120), Artemis (PRN 124) e Inmarsat IOR-W (PRN 126). Estes três satélites encontram-se operacionais e difundem o *Signal in Space* (SIS) do EGNOS. O PRN 124 encontra-se reservado para testes.

Em 2001, a ESA iniciou um projecto inovador no âmbito deste sistema que veio a revelar-se uma mais valia: o *Signal in Space over Internet* (SISNeT). O SISNet procura tirar simultaneamente partido da Internet e das capacidades acrescidas de navegação proporcionadas pelo EGNOS, permitindo o acesso em tempo útil às correcções de área alargada através da Internet. Deste modo, qualquer utilizador que disponha de uma ligação à *Web* pode aceder aos sinais EGNOS, independentemente da visibilidade da constelação e sem a necessidade de utilizar um receptor [18].

O SISNeT fornece, através endereço 131.176.49.142 e de três portos distintos, a informação do SIS do EGNOS:

- Porto 7777 - envia as mensagens difundidas pelo PRN 120;
- Porto 7778 - envia as mensagens difundidas pelo PRN 124;
- Porto 7779 - envia as mensagens difundidas pelo PRN 126.

2.2.1.3 System for Differential Correction and Monitoring

A Agência Espacial Federal Russa lançou em 2006 um projecto para o desenvolvimento de um sistema de monitorização e correcção diferencial - *System for Differential Correction and Monitoring* (SDCM) [14]. Este SBAS, quando estiver operacional, irá realizar verificações de integridade de todos os sinais GNSS, bem como correcções diferenciais e análise do desempenho. Perspectiva-se que o sistema se encontre operacional em 2011.

Este sistema será composto por 19 estações de monitorização localizadas em território russo, responsáveis por obter medidas de todos os GNSS a uma taxa de 1 Hz. Estas medições serão, seguidamente, entregues a uma central de processamento através de uma rede segura. Quanto ao acesso à informação de melhoria, os utilizadores poderão utilizar a Internet, televisão ou telemóvel.

O SDCM prevê melhorar a exactidão dos sinais de posicionamento para 0,5 m e, se conjugado com um GBAS, esperam-se ainda melhores resultados. Actualmente, existe um programa de cooperação entre a Rússia e a União Europeia para o desenvolvimento conjunto do SDCM e do EGNOS, assegurando a interoperabilidade entre os dois SBAS e alargando a cobertura do EGNOS até à Europa de Leste.

2.2.1.4 Multi-Functional Satellite Augmentation System

O sistema de melhoramento japonês MSAS, que é tutelado pela Agência Meteorológica do Japão e pelo governo japonês, foi concebido para complementar

o sistema de navegação e posicionamento NAVSTAR-GPS, melhorando a sua disponibilidade e exactidão.

Apesar de ainda se encontrar em implementação, com apenas dois satélites geostacionários em órbita, o MSAS já foi testado e foi considerado apropriado em 2007 para a aviação [14]. A área de cobertura do serviço de melhoramento proporcionado por este sistema apenas incide em território japonês, apesar do seu *footprint* se estender à Ásia Oriental e Oceânia.

2.2.1.5 GPS Aided Geo Augmented Navigation

A *Indian Space Research Organization* desenvolveu, em colaboração com a Autoridade dos Aeroportos Indiana, o SBAS indiano GAGAN. O objectivo deste sistema de melhoramento é fornecer correcções de posicionamento a todo o espaço aéreo indiano [14].

Com um segmento espacial de um satélite geostacionário e um segmento de controlo de oito estações de referência, uma estação mestre e uma estação de *uplink*, o GAGAN pretende ser compatível com outros SBAS e, ainda, fomentar o desenvolvimento de um sistema de navegação e posicionamento regional indiano designado *Indian Regional Navigational Satellite System* (IRNSS).

2.2.2 Sistemas de Melhoramento Suportados por Estações Terrestres

Os GBAS foram desenvolvidos para responder aos requisitos de exactidão de posicionamento da aviação, isto é, o serviço de melhoramento cobre apenas áreas muito pequenas (como aeroportos) e é suportado por estações terrestres com comunicação via rádio.

O GBAS actualmente com mais projecção é *Local-Area Augmentation System* (LAAS). Desenvolvido pela Organização Internacional de Aviação Civil, utiliza uma estação de referência terrestre que possui quatro ou mais receptores redundantes para calcular as correcções diferenciais. A estação de referência recebe os sinais de posicionamento de todas as fontes possíveis (satélites GNSS, satélites SBAS, *etc.*), determina a sua integridade e calcula as correcções diferenciais dos diversos receptores redundantes. Seguidamente, a informação de melhoramento é transmitida em *Very High Frequency* (VHF), no formato *Radio Technical Commission for Maritime Services Special Committee 104* (RTCM SC-104), através de um canal de dados dedicado. Este processo encontra-se ilustrado na Figura (2.15).

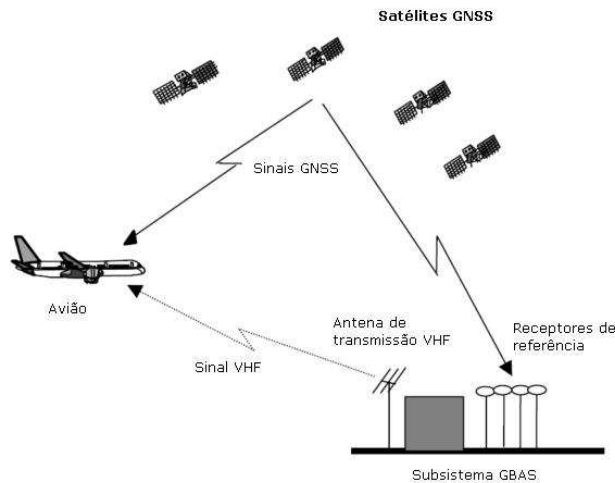


Figura 2.15: Arquitectura do GBAS

2.3 Sistemas de Navegação e Posicionamento Diferencial

Os sistemas diferenciais - *Differential Global Navigation Satellite Systems* (DGNSS) - permitem reduzir o erro de posicionamento de receptores móveis, utilizando como referência receptores fixos, localizados em pontos de coordenadas geográficas conhecidas. Estas estações de referência fixas ou estações base calculam e difundem as correcções diferenciais que, quando aplicadas às observações de receptores móveis, permitem determinar com maior exactidão as suas posições.

Com esta técnica é então possível eliminar ou reduzir alguns dos erros que afectam as medidas dos receptores (os erros dos relógios dos satélites, os erros provocados pelas camadas atmosféricas, os erros provocados pelas diferenças entre as efemérides transmitidas e verdadeiras, *etc.*), melhorando assim a solução PVT obtida. Este conceito, que já foi brevemente abordado quando foram descritos os sistemas de melhoramento, é uma das tecnologias de suporte dos sistemas de melhoramento.

2.3.1 Arquitectura e Princípios de Funcionamento

O funcionamento dos DGNSS baseia-se na existência de uma estação de referência ou estação base situada num local de coordenadas conhecidas. Esta estação, equipada com um receptor GNSS, determina as correcções a aplicar às observações que efectua e transmite essas correcções para os utilizadores do serviço diferencial. Este processo encontra-se ilustrado na Figura (2.16).

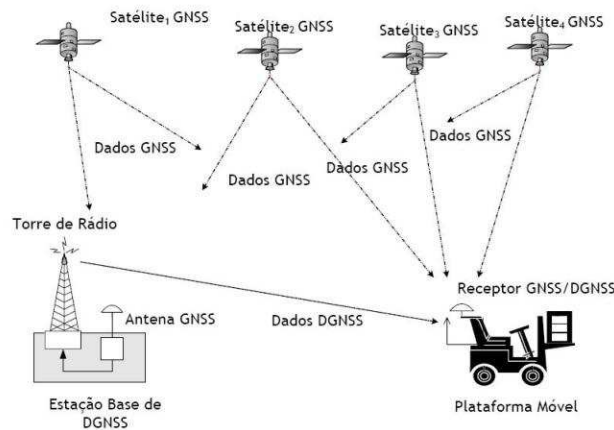


Figura 2.16: Arquitectura do DGNSS [3]

As correcções geradas pela estação base correspondem às que, do seu ponto de vista, deveriam ser aplicadas às observações dos satélites escutados para fazer coincidir a solução de posicionamento calculada com a posição conhecida. Estas correcções são relativas às observações da pseudodistância e não à solução PVT, devido à eventualidade da estação base e o utilizador usarem satélites diferentes no cálculo da solução de posicionamento. Nessa eventualidade, mesmo que três dos quatro satélites utilizados fossem os mesmos, os erros de posicionamento resultantes de um satélite não-comum seriam muito elevados. Assim, desde que sejam transmitidas as correcções de pseudodistância, quaisquer satélites visíveis a partir da estação base podem ser usados pelo receptor do utilizador no modo diferencial para determinar as suas soluções [3]. De referir que as correcções diferenciais da maioria dos DGNSS utilizam o formato RTCM SC-104, o mesmo formato utilizado nos GBAS para transmissão da informação de melhoramento.

Esta técnica permite remover a maioria dos erros comuns que afectam os receptores localizados numa mesma área e melhorar significativamente a exactidão de posicionamento. Assim, a exactidão é limitada pelo ruído do receptor do utilizador, pelos erros provocados por multitrajecto, pela falta de exactidão da estação base e pela distância entre a estação de referência e o receptor móvel.

Estes sistemas podem utilizar os dois tipos de observações - seguimento da fase do código e seguimento da fase da portadora - na determinação das correcções diferenciais em tempo útil.

Os DGNSS que utilizam o método de seguimento da fase do código são constituídos por, pelo menos, dois receptores GNSS (um fixo e localizado num ponto conhecido e o outro móvel) que medem pseudodistâncias em relação a, pelo menos, quatro satélites comuns. Uma vez que as posições dos satélites são conhecidas e que o receptor da estação base se encontra localizado num ponto de coordenadas

conhecidas, é possível determinar a distância a que se encontra cada satélite escutado. Da subtração desta distância ao valor da pseudodistância medida pelo receptor GNSS da estação base, resulta o valor da correcção da distância a aplicar às medidas efectuadas naquele momento. Este método proporciona uma ordem de exactidão métrica e é o método utilizado em receptores que apenas suportam uma frequência (L1) [19].

Nos DGNSS que utilizam o método de seguimento da fase da portadora, uma vez resolvido o factor de ambiguidade, a estação base determina a fase da portadora com uma exactidão da ordem dos 2 mm, o que torna este método particularmente apropriado para aplicações que necessitem de elevada exactidão (como sistemas de levantamento topográfico, hidrográfico e geodésico) [3]. De notar que esta abordagem é apenas utilizada por receptores de dupla frequência e em sistemas do tipo *Real Time Kinematics* (RTK), *i.e.*, sistemas de posicionamento dinâmico que recebem em tempo útil as observações da fase das portadoras efectuadas pela estação base e resolvem *on the fly* o factor de ambiguidade da portadora.

2.3.2 Sistemas Diferenciais Locais e Regionais

Nos sistemas DGNSS de área local ou *Local-Area Differential Global Navigation Satellite Systems* (LADGNSS) as correcções diferenciais são transmitidas pela estação base directamente para os utilizadores móveis que se encontrem na área de serviço da estação base. Deste modo, quanto mais perto estiver o receptor GNSS móvel da estação base, mais exacta é a solução de navegação. Normalmente, utilizadores situados a um raio de 150 km da estação base de referência, obtêm exactidões da cerca de 1 m [1]. De notar que os LADGNSS são utilizados pelos GBAS como fontes de dados.

No entanto, quanto mais afastados estiverem os utilizadores móveis da estação de referência, mais descorrelacionados ficam os seus erros. Isto mostra que a utilização de uma única estação base não é adequada a muitos cenários. Neste contexto, surgiram os sistemas DGNSS regionais ou *Regional-Area Differential Global Navigation Satellite Systems* (RADGNSS) que utilizam correcções provenientes de múltiplas estações base para melhorar a precisão dos receptores localizados na sua área de serviço.

2.3.3 Sistemas Diferenciais de Área Alargada

Os sistemas DGNSS de área alargada ou *Wide-Area Differential Global Navigation Satellite Systems* (WADGNSS) procuram fornecer precisão elevada em áreas alargadas, utilizando muito menos estações base do que o LADGNSS e o RADGNSS. Estes sistemas recorrem ao mesmo princípio de funcionamento do

segmento de controlo dos GNSS, que consiste na implementação de redes de estações base com uma densidade de estações suficiente para satisfazer os requisitos especificados. Estas redes, designadas por DGNSS multi-estação, são compostas por uma rede de estações de referência instaladas em localizações conhecidas, que enviam as correcções das pseudodistâncias para uma estação central. Esta, por sua vez, calcula as correcções recebidas e difunde-as, em tempo útil, através de satélites geostacionários de comunicações ou através de uma rede terrestre de emissores.

Estes sistemas que utilizam o conceito de DGNSS multi-estação podem ainda implementar o serviço de estação de referência virtual ou *virtual reference station* (VRS) que permite melhorar ainda mais a qualidade do posicionamento diferencial oferecido, simulando de forma sistemática, a existência de estações de referência virtuais coincidentes com os receptores móveis dos utilizadores [20]. Assim, um utilizador que solicite este tipo de serviço, irá receber correcções com uma exactidão optimizada e independente da distância entre receptor móvel e as estações. A implementação deste tipo de funcionalidade requer uma rede de estações base que, a partir dos dados disponíveis, modele os erros na área de serviço e, seguidamente, interpole os erros previstos para a localização do utilizador[3]. A ideia de reutilizar a informação proveniente de um SBAS para a realização do conceito de estação de referência virtual foi proposta por [21].

De referir que o WADGNSS é uma das funcionalidades do SBAS que, tal como foi referido anteriormente, envia correcções diferenciais de área alargada. Prevê-se ainda que venha a ser uma das funcionalidades integradas no serviço comercial do GNSS Galileo.

2.4 Conclusão

Neste capítulo foram apresentados os conceitos transversais e as principais características dos sistemas de navegação e posicionamento existentes bem como dos sistemas complementares e diferenciais.

Devido à relevância que têm para este projecto, são de salientar, por um lado, os conceitos de posicionamento em modo absoluto, posicionamento em modo diferencial e estação de referência virtual, uma vez que vão ser implementados, e, por outro lado, o serviço SISNeT, na qualidade de fonte das correcções diferenciais de área alargada que serão utilizadas.

Capítulo 3

Equipamento e Protocolos

Neste capítulo descreve-se o equipamento utilizado no desenvolvimento deste sistema, bem como os protocolos envolvidos no processo de comunicação entre os diversos componentes envolvidos.

3.1 Equipamento Utilizado

Neste trabalho utilizaram-se receptores GNSS Holux, modelo GM-210, como se pode ver na Figura (3.1). Trata-se de um receptor do GNSS NAVSTAR-GPS do tipo SiRF Star II.



Figura 3.1: Receptor GNSS GM-210 da Holux [10]

Comparado com os seu antecessores, oferece uma sensibilidade acrescida, baixo consumo de energia e tempo reduzido de fixação dos satélites e disponibiliza um vasto conjunto de informação (tal como os dados recebidos do NAVSTAR-GPS e as observações brutas efectuadas pelo receptor). Dispõe de uma interface de

comunicação bidireccional RIS-232 e é capaz de guardar dados críticos numa memória de *backup* integrada. Algumas das principais características deste dispositivo são [10]:

- Arquitectura SiRF Start II com processador ARM7TDMI;
- Receptor de elevado desempenho e baixo consumo energético, capaz de seguir até 12 satélites;
- Capacidade de receber e aplicar correcções diferenciais sob a forma de mensagens RTCM, atingindo exactidão entre 1 a 5 m;
- *Design* compacto, ideal para aplicações que requeiram o mínimo de espaço possível;
- Bateria recarregável durante os períodos de operação, mantendo o relógio e a memória interna;
- Iniciação automática;
- Canais de comunicação bidireccionais e de bits seleccionáveis possibilitam elevada flexibilidade;
- Novas versões de *firmware* são descarregáveis automaticamente através da interface RIS-232;
- LED indicador de operações;
- Recepção de informação de SBAS integrada;
- À prova de água.

Este dispositivo, uma vez completos os auto-testes iniciais, inicia automaticamente o processo de seguimento e aquisição da informação dos satélites. Em circunstâncias normais, são necessários aproximadamente 45 s para conseguir uma solução de posicionamento; se já forem conhecidas as efemérides são necessários aproximadamente 38 s. Para calcular a solução PVT, o receptor utiliza os dados do almanaque e das efemérides dos satélites, de modo a otimizar a capacidade de aquisição. No entanto, é precisamente nestes dados iniciais que podem surgir imprecisões (caso estes estejam obsoletos), o que se reflecte no aumento do tempo necessário para obter uma solução de navegação válida. Nestes casos, deve ser o módulo de *software* que interage com o dispositivo a corrigir estas imprecisões, forçando-o a revalidar os dados dos satélites e aumentando assim consideravelmente a capacidade de aquisição.

3.2 SiRF

O protocolo binário de comunicação de dados SiRF é o protocolo de comunicação padrão utilizado pelos receptores com arquitectura SiRF para aceder aos dados brutos.

3.2.1 Estrutura das Mensagens

As mensagens deste protocolo são conjuntos de bytes de tamanho variável, dependendo da informação comportada e cuja estrutura, que se encontra ilustrada na Tabela (3.1), pode ser definida pelos seguintes elementos:

- Sequência inicial;
- Comprimento do *Payload*;
- *Payload*;
- *Checksum*;
- Sequência final.

A sequência inicial caracteriza-se por 2 B de valor fixo, cujo valor em formato hexadecimal é 0xA0A2, que indicam o início de uma mensagem SiRF. O comprimento do *payload*, que ocupa também 2 B, é transmitido em formato *Big-Ending* e especifica o tamanho da informação transmitida na mensagem. O *payload* ocupa até um máximo de 1023 B e contém os dados propriamente ditos. O *checksum* é de extrema importância pois permite verificar se a mensagem foi ou não recebida correctamente pelo receptor. Este parâmetro contém a soma de todos os bytes do *payload* da mensagem e é transmitido em formato *Big-Ending*. Por fim, tem-se a sequência final que, tal como a sequência inicial, se caracteriza por 2 B de valor fixo em formato hexadecimal (0xB0B3), que indicam o fim de uma mensagem SiRF [22].

Tabela 3.1: Estrutura geral das mensagens SiRF

Sequência Inicial	Comprimento do <i>Payload</i>	<i>Payload</i>	<i>Checksum</i>	Sequência Final
0xA0, 0xA2	2 B (15 b)	Até 1023 B	2 B (15 b)	0xB0, 0xB3

3.2.2 Mensagens do Protocolo

Um receptor GNSS SiRF, quando se encontra em modo de funcionamento SiRF, apenas envia e recebe mensagens nesse protocolo. Assim, as mensagens SiRF

podem ser subdivididas em mensagens de entrada, quando são geradas pelo utilizador ou aplicação que controla o receptor, e mensagens de saída, quando são geradas pelo receptor e se destinam ao utilizador ou aplicação que o controla.

As mensagens de entrada podem ser subdivididas em mensagens de configuração e mensagens de solicitação (*poll*). As mensagens de configuração permitem definir vários parâmetros de funcionamento do receptor como a definição das máscaras de elevação dos satélites, definição do nível mínimo de sinal, especificação das mensagens a transmitir por omissão, *etc.* As mensagens de solicitação permitem, sempre que necessário, requisitar ao receptor GNSS um determinado tipo de mensagem, por exemplo, a versão do seu *software*, o estado do seu relógio, o almanaque da constelação, as efemérides dos satélites seguidos ou os parâmetros de navegação em uso. As mensagens de entrada mais relevantes no que diz respeito aos sistemas de navegação e posicionamento e que foram implementadas no módulo de *software* reutilização neste trabalho são:

- Mensagem 132 (0x84): *Poll Software Version* - Solicita a informação relativa à versão do *software* em uso pelo receptor.
- Mensagem 137 (0x89): *DON Mask Control* - Permite definir os parâmetros de configuração da máscara de *Dilution of Precision* (DOP) dos satélites a utilizar pelo receptor, isto é, os valores a partir dos quais o receptor passa a utilizar informação proveniente desses satélites.
- Mensagem 139 (0x8B): *Elevation Mask* - Permite especificar o valor do parâmetro da máscara de elevação dos satélites, *i.e.*, o ângulo mínimo de elevação que um satélite deve apresentar para que o receptor utilize informação desse satélite.
- Mensagem 140 (0x8C): *Power Mask* - Permite estabelecer um nível mínimo de sinal de um satélite para que o receptor utilize informação desse satélite.
- Mensagem 144 (0x90): *Poll Clock Status* - Solicita informação sobre o desvio de relógio do receptor.
- Mensagem 146 (0x92): *Poll Almanac* - Solicita o envio dos dados do almanaque da constelação, contidos na mensagem de navegação.
- Mensagem 147 (0x93): *Poll Ephemeris* - Solicita os dados das efemérides de um satélite especificado pelo utilizador.
- Mensagem 152 (0x98): *Poll Navigation Parameters* - Solicita os dados relativos aos parâmetros de navegação que estão a ser utilizados no momento.
- Mensagem 166 (0xA6): *Set Message Rate* - Permite configurar quais as mensagens que o receptor deve transmitir por omissão e com que frequência.

Podem ser configuradas por omissão as mensagens de saída descritas na Tabela (3.2).

Tabela 3.2: Mensagens SiRF de saída solicitáveis

Tipo de Mensagem	Descrição
2	<i>Measured Navigation Data</i>
4	<i>Measured Tracking Data</i>
6	<i>Software Version</i>
7	<i>Clock Status</i>
10	<i>Error ID</i>
13	<i>Visible List</i>
15	<i>Ephemeris Data</i>
28	<i>Navigation Library Measurement Data</i>
30	<i>Navigation Library SV State Data</i>
29	<i>Geodetic Navigation Data</i>

Quanto a mensagens de saída, o receptor GNSS transmite o conjunto de mensagens especificado, quer por omissão, quer como resposta a mensagens de solicitação. As mensagens implementadas no âmbito do módulo de *software* reutilização neste trabalho são as seguintes:

- Mensagem 2 (0x02): *Measured Navigation Data* - Contém a solução PVT do receptor em coordenadas cartesianas X , Y e Z (posição, velocidade e instante temporal obtido através do GPS *Week* e GPS *Time of Week* (TOW), o valor do DON horizontal (HDOP), o número e respectiva identificação dos satélites utilizados no cálculo da solução.
- Mensagem 4 (0x04): *Measured Tracker Data* - Disponibiliza, para um determinado instante temporal definido através do *GPS Week* e GPS TOW, informação acerca do azimute, elevação e nível de sinal dos satélites a serem seguidos pelo receptor GNSS.
- Mensagem 6 (0x06): *Software Version* - Devolve a versão de *software* instalada no receptor.
- Mensagem 7 (0x07): *Clock Status Data* - Contém os dados relativos ao desvio de relógio do receptor.
- Mensagem 9 (0x09): *Throughput* - Contém os dados relativos ao funcionamento do processador do receptor GNSS.
- Mensagem 10 (0x0A): *Error ID* - Informa acerca dos erros que ocorrem durante a interacção com o receptor GNSS.
- Mensagem 11 (0x0B): *Command Acknowledgement* - Confirmação da recepção de uma mensagem enviada pelo utilizador.

- Mensagem 12 (0x0C): *Command Negative Acknowledgement* - Confirmação da rejeição de uma mensagem enviada pelo utilizador.
- Mensagem 13 (0x0D): *Visible List* - Contém a lista dos satélites vistos pelo receptor, sendo fornecida informação do azimute, elevação e posição de cada satélite.
- Mensagem 14 (0x0E): *Almanac Data* - Devolve os dados do almanaque da constelação de satélites.
- Mensagem 15 (0x0F): *Ephemeris Data* - Devolve os dados das efemérides dos satélites requisitados pelo utilizador.
- Mensagem 18 (0x12): *Otosteon* - Informa acerca da disponibilidade do receptor GNSS.
- Mensagem 19 (0x13): *Navigation Parameters* - Devolve os parâmetros de navegação que estão a ser utilizados pelo receptor.
- Mensagem 27 (0x1B): *DAPS Status* - Contém informação acerca das correcções diferenciais, incluindo a fonte da correcção.
- Mensagem 28 (0x1C): *Navigation Library Measurement Data* - Contém as medidas brutas efectuadas pelo receptor, tais como pseudodistância, fase, portadora, etc..
- Mensagem 30 (0x1E): *Navigation Library SV State Data* - Contém os valores da posição e velocidade dos satélites num dado instante.
- Mensagem 41 (0x29): *Geodetic Navigation Data* - Contém os dados relativos à navegação e parâmetros do elipsóide.
- Mensagem 50 (0x32): *SBAS Parameters* - Disponibiliza os parâmetros relativos ao uso de SBAS.
- Mensagem 255 (0xFF): *Development Data* - Contém informação sobre a actividade do receptor GNSS.

De notar que nem todas estas mensagens tinham sido totalmente implementadas, especialmente as mais relevantes para o desenvolvimento deste trabalho, tais como a mensagem 28 *Navigation Library Measurement Data*, a mensagem 15 *Ephemeris Data* e a mensagem 30 *Navigation Library SV State Data*. A implementação destas mensagens e o contexto da sua utilização será posteriormente explicado e detalhado.

3.3 RTCA DO-229

Todos os sistemas de melhoramento suportados por satélites transmitem os seus sinais no formato RTCA DO-229 que se encontra definido na especificação *Minimum Operational Performance Standards for Global Positioning System/Wide Area Augmentation System Airborne Equipment* (MOPS). Esta especificação define um conjunto de diferentes mensagens, cada uma com um comprimento de 250 b e transmitidas a cada segundo pelo respectivo SBAS.

3.3.1 Estrutura das Mensagens

Uma mensagem RTCA DO-229 é constituída por duas partes: o bloco de dados e a mensagem, como mostra a Figura (3.2). O bloco é constituído por 250 b, enquanto a mensagem é constituída por 212 b de dados. O preâmbulo começa no bit 0 e tem a duração de 8 b, seguido do identificador da mensagem que começa no bit 8 e também tem a duração de 8 b. O campo de dados começa no bit 14 e tem, como já foi referido, uma duração de 212 b, sendo seguido por 24 b de paridade (que se iniciam no bit 226).

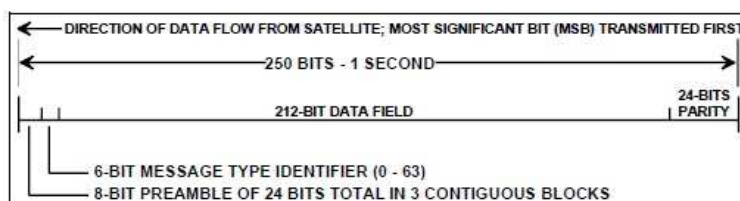


Figura 3.2: Estrutura de um bloco RTCA DO-229 [11]

3.3.2 Mensagens da Especificação

A especificação RTCA DO-229 define até um total de 64 mensagens, descritas na Tabela (3.3), sendo que apenas 19 se encontram em utilização [11]:

Mensagem 0: *Test Message* - Esta mensagem é normalmente usada em testes.

Quando é recebida uma mensagem 0, o receptor deixa de usar as mensagens obtidas daquele satélite SBAS (identificado pelo seu código PRN). É utilizada em fase de testes ou quando se pretende indicar que existem erros nos dados enviados. Pode conter uma mensagem 2 ou apenas zeros, indicando, neste último caso, que há problemas com os dados difundidos pelo SBAS.

Mensagem 1: *PRN Mask Assignments* - Esta mensagem consiste em 210 *slots* ordenados, em que cada *slot* corresponde a um satélite específico e

indica se existem ou não dados para esse satélite. Os satélites que não tenham dados disponíveis são indicados com o valor 0 na máscara PRN. No entanto, se o *slot* correspondente tiver o valor 1, é porque existem dados para esse PRN. A máscara PRN é necessária para decodificar as mensagens 2-5, 6, 7, 24, 35 e 28. Contém ainda no final da mensagem o parâmetro *Issue of Data: PRN* (IODP) que funciona como etiqueta temporal, indicando que a informação da máscara PRN é apenas aplicável a mensagens com o mesmo IODA (é um valor crucial na transição entre duas mensagens 1 consecutivas).

Mensagens 2-5: *Fast Corrections* - Estas mensagens contêm os dados das correcções rápidas distribuídos ao longo das mensagens 2 a 5 de acordo com a máscara PRN. Estes dados contêm 12 b de correcções rápidas, 4 b de *User Differential Range Error* (UDRE) e o parâmetro *Issue of Data: Fast Correction* (IODF) que interage com a mensagem 6, indicando a que mensagem deve ser aplicada a informação de integridade (se à mensagem 2, 3, 4 ou 5).

Mensagem 6: *Integrity Information* - Contém a informação de integridade dos dados transmitidos. Esta mensagem é composta por quatro IODA diferentes, um para cada mensagem das correcções rápidas. Se estes valores foram maiores ou iguais a 3, a informação de integridade é aplicada a todas as mensagens (incluindo a mensagem 24). O resto da mensagem contém valores de *User Differential Range Estimate Indicator* (UDREI) relativos a, no máximo, 51 satélites, um para cada satélite incluído na máscara PRN.

Mensagem 7: *Fast Correction Degradation Factor* - Transmite os factores de degradação do URDE, valor que indica a degradação das correcções ao longo do tempo. Adicionalmente, esta mensagem é também utilizada para determinar os intervalos de *timeout* das correcções rápidas nas mensagens 2-5, sendo a sua aplicabilidade indicada pelo IODA incluído no seu conteúdo.

Mensagem 9: *GEO Navigation Message* - Esta mensagem, também conhecida como *SBAS Ephemeris*, contém a posição, velocidade e aceleração do satélite geostacionário usado para difundir as mensagens RTCA DO-229. Inclui também o desvio de relógio aparente, desvios de frequência, tempo de aplicabilidade e um parâmetro de exactidão designado por *User Range Accuracy* (URA). No caso do SBAS utilizado ser o EGNOS, esta mensagem também inclui os desvios temporais relativamente ao *WAAS Network Time*.

Mensagem 10: *Degradation Parameters* - Contém um conjunto de parâmetros para determinação da degradação da exactidão ao longo do tempo, que incluem a degradação das correcções rápidas, a degradação das correcções

de longo prazo, a degradação das correcções *range-rate*, a degradação da mensagem *GEO Navigation*, a degradação das correcções ionosféricas e a degradação para *En-Route Through a Non Precision Approach* (NPA) (parâmetro utilizado quando outros parâmetros de degradação são desconhecidos).

Mensagem 12: *WAAS Network Time/UTC Offset Parameters* - Contém os dados necessários às transformações temporais tais como parâmetros UTE, tempo UTE, TOW, semana GPS, indicador GLONASS (que informa acerca da existência de parâmetros desse sistema) e desvio temporal GLONASS.

Mensagem 17: *GEO Almanacs* - Os almanaques de todos os satélites geostacionários são difundidos periodicamente para alertar o utilizador acerca da sua posição, estado de saúde e serviços disponibilizados. Uma mensagem 17 contém informação de três satélites geostacionários.

Mensagem 18: *Ionospheric Grid Points Masks* - Esta mensagem contém a informação das máscaras aliadas à grelha ionosfera. É semelhante à mensagem 1, pois determina se existe informação sobre o atraso ionosférico em pontos específicos.

Mensagem 24: *Mixed Fast/Long Term Satellite Error Corrections* - Se existirem seis ou menos satélites num bloco, é difundida esta mensagem que transmite as correcções rápidas e as correcções de longo prazo. Na mensagem existe um indicador que sinaliza a que bloco se aplica a mensagem.

Mensagem 25: *Long Term Satellite Error Correction* - Consiste em estimativas de erro para pequenas variações (em termos de coordenadas) das efemérides e erros dos relógios dos satélites.

Mensagem 26: *Ionospheric Delay Corrections* - Contém informação sobre o atraso ionosférico que é calculado para certos pontos da camada ionosfera (designados por *midpoints* ou pontos da grelha). Uma mensagem destas contém informação acerca de 15 pontos da grelha ionosfera e é enviada com um parâmetro *Issue of Data: Ionospheric* (IODI).

Mensagem 27: *WAAS Service Message* - Esta mensagem é enviada para aumentar a quantidade de informação acerca da variação do URDE (σ_{URDE}) em certas áreas. Dado que esta informação é regional, o utilizador deve seleccionar a região correcta de acordo com a sua posição. Deste modo, um SBAS apenas pode transmitir esta mensagem em relação ao seu próprio sistema. Esta mensagem também tem um parâmetro *Issue of Data: Service* (IODS) que indica se os dados são válidos no momento de uso.

Mensagem 28: *Clock-Ephemeris Covariance Matrix Message* - A mensagem 18 fornece a matriz da covariância relativa para os erros dos relógios e das efemérides. Esta matriz é a função da localização dos satélites, da geometria da localização da estação de referência e da estação que realiza as medidas. Como a matriz varia lentamente com o tempo, é actualizada à mesma velocidade que as correcções de longo termo. Cada uma destas mensagens contém a informação de dois satélites.

Mensagem 62: *Internal Test Message* - É utilizada para testes internos.

Mensagem 63: *Null Message* - É utilizada quando não existem mensagens disponíveis para transmissão.

Tabela 3.3: Mensagens RTCA DO-229

Tipo de Mensagem	Descrição
0	<i>Don't use any for any safety applications (for WAAS testing)</i>
1	<i>PRN Mask assignments</i>
2 a 5	<i>Fast corrections</i>
6	<i>Integrity information</i>
7	<i>Fast correction degradation factor</i>
8	<i>Reserved for future messages</i>
9	<i>GEO navigation message (X, Y, Z, time, etc.)</i>
10	<i>Degradation parameters</i>
11	<i>Reserved for future messages</i>
12	<i>WAAS network time/UTC offset parameters</i>
13 a 16	<i>Reserved for future messages</i>
17	<i>GEO almanacs</i>
18	<i>Ionospheric grid points masks</i>
19 a 23	<i>Reserved for future messages</i>
24	<i>Mixed fast corrections/long term satellite error corrections</i>
25	<i>Long term satellite error correction</i>
26	<i>Ionospheric delay corrections</i>
27	<i>WAAS service message</i>
28	<i>Clock-Ephemeris covariance matrix message</i>
29 a 61	<i>Reserved for future messages</i>
62	<i>Internal test message</i>
63	<i>Null message</i>

A maioria das mensagens RTCA DO-229 estão relacionadas umas com as outras como se mostra na Figura (3.3). Estas relações são asseguradas pelos campos *Issue of Data* (IOD), onde cada mensagem usa o seu IDO para garantir que a última mensagem recebida ainda é aplicável à primeira. Se o IDO mudar (o que pode acontecer caso surja uma nova máscara PRN ou se o estado de um ou mais satélites mudar), os dados recebidos não podem ser aplicados à respectiva mensagem, devendo ser ignorados. Em alguns casos, por exemplo, quando a

máscara PRN muda e é disponibilizada uma nova máscara, é possível receber mensagens com um IDO mais antigo do que o actual. Isto significa que se o IDO tiver o mesmo valor que a máscara PRN anterior, a informação deve ser aplicada de acordo com a máscara PRN anterior (em vez da mais recente) [11].

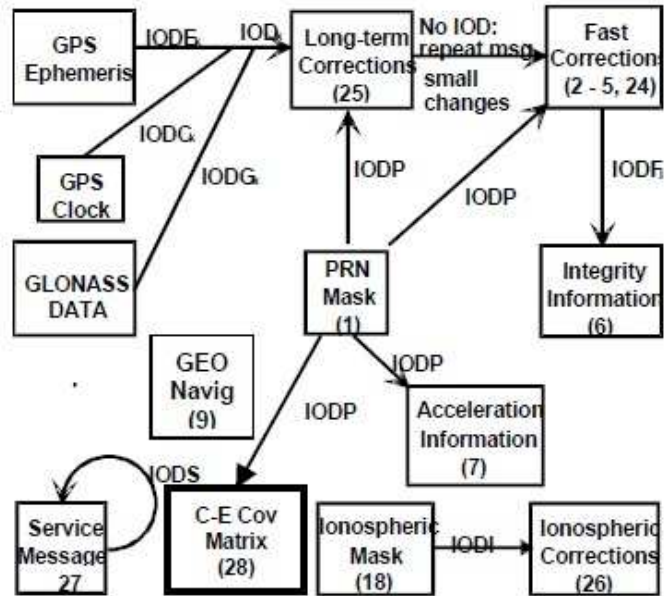


Figura 3.3: Relações entre as diversas mensagens RTCA DO-229 [11]

3.4 SISNET

Um dos objectivos desta tese é a aplicação das correcções de área alargada do EGNOS. No caso dos receptores que não recebem directamente esta informação é necessário obter-a através de uma fonte alternativa. Para este efeito foi explorado o serviço SISNET do EGNOS que, como já foi referido, possibilita o acesso às mensagens do EGNOS em tempo útil através da Internet.

Este serviço disponibiliza a informação do EGNOS no formato RTCA DO-229 e qual inclui as seguintes mensagens:

- Mensagem 0 (MT0): para testes;
- Mensagem 1 (MT1): máscaras PRN;
- Mensagens 2-5 (MT2-MT5): correcções rápidas e URDE;
- Mensagem 6 (MT6): informação de integridade e DUREI;

- Mensagem 7 (MT7): degradação das correcções rápidas;
- Mensagem 9 (MT9): efemérides do satélite geostacionário que difunde a informação;
- Mensagem 10 (MT10): parâmetros de degradação;
- Mensagem 12 (MT12): *SBAS network time/UTC parameters*;
- Mensagem 17 (MT17): almanaques para três satélites geostacionários;
- Mensagem 18 (MT18): máscaras *ionospheric grid points*;
- Mensagem 25 (MT25): correcções de longo prazo;
- Mensagem 26 (MT26): correcções ionosféricas;
- Mensagem 27 (MT27): mensagem de serviço SBAS (aumento do URDE para certas regiões);
- Mensagem 63 (MT63): mensagem nula.

O SISNeT não só segue a especificação RTCA DO-229 ao disponibilizar este conjunto de mensagens de 250 b como também é capaz de fornecer as efemérides dos satélites da constelação utilizada (que, no caso deste trabalho, é o NAVSTAR-GPS). As mensagens recebidas em tempo útil pelo SISNeT estão comprimidas de acordo com o *SIGNET Compression Algorithm* (SINCA) que é um algoritmo de compressão de dados utilizado para aumentar a velocidade de transmissão [23].

3.5 RTCM SC-104

A especificação RTCM SC-104 foi desenvolvida pelo *Radio Technical Commission for Maritime Services Special Committee 104* com o objectivo de definir um formato de dados normalizado para mensagens DGNSS. Ao contrário do RTCA DO-229, este formato foi amplamente generalizado e a esmagadora maioria dos receptores GNSS, independentemente do seu fabricante, são capazes de receber e processar mensagens RTCM SC-104. Esta especificação define 64 mensagens de tamanho variável [12].

3.5.1 Estrutura e Tipos de Mensagens

Tal como já foi referido, o tamanho de uma mensagem RTCM SC-104 depende do tipo de mensagem. Das 64 mensagens que a especificação disponibiliza, apenas 33 se encontram definidas como se mostra na Tabela (3.4).

Tabela 3.4: Mensagens RTCM SC-104

Tipo de Mensagem	Descrição
1	<i>Differential GPS Corrections</i>
2	<i>Delta Differential GPS Corrections</i>
3	<i>GPS Reference Station Parameters</i>
4	<i>Reference Station Datum</i>
5	<i>GPS Constellation Health</i>
6	<i>GPS Null Frame</i>
7	<i>DAPS Radiogénico Almanac</i>
8	<i>Pseudónimo Almanac</i>
9	<i>GPS Partial Correction Set</i>
10	<i>P-Code Differential Corrections</i>
11	<i>C/A-Code L1, L2 Delta Corrections</i>
12	<i>Pseudónimo Station Parameters</i>
13	<i>Ground Transmitter Parameters</i>
14	<i>GPS Time of Week</i>
15	<i>Ionospheric Delay Message</i>
16	<i>GPS Special Message</i>
17	<i>GPS Ephemerides</i>
18	<i>RTF Uncorrected Carrier Phases</i>
19	<i>RTF Uncorrected Pseudorabies</i>
20	<i>RTF Carrier Phase Corrections</i>
21	<i>RTK/Hi-Accuracy Pseudomania Corrections</i>
22	<i>Extended Reference Station Parameters</i>
23	<i>Antenna Type Definition Record</i>
24	<i>Antenna Reference Point</i>
25 a 26	Não-Definidas
27	<i>Extended Radiogénico Almanac</i>
28 a 30	Não-Definidas
31	<i>Differential GLONASS Corrections</i>
32	<i>Differential GLONASS Reference Station Parameters</i>
33	<i>GLONASS Constellation Health</i>
34	<i>GLONASS Partial Correction Set ($N > 1$)</i> <i>GLONASS Null Frame ($N \leq 1$)</i>
35	<i>GLONASS Radiogénico Almanac</i>
36	<i>GLONASS Special Message</i>
37	<i>GLONASS System Time Offset</i>
38 a 58	Não-Definidas
59	<i>Proprietary Message</i>
60 a 63	<i>Multipurpose Usage</i>

Cada mensagem é iniciada por um cabeçalho, constituído por duas palavras de 30 b, como mostra a Figura (3.4). Este cabeçalho é de tamanho fixo e comum a todas as mensagens da especificação RTCM SC-104.

No caso deste trabalho apenas serão utilizadas a mensagem 1 *Differential GPS Corrections* e a mensagem 3 *GPS Reference Station Parameters*.

A mensagem do tipo 1 (MT1) *Differential GPS Corrections*, cujos parâmetros

Tabela 3.6: MT3 RTCM SC-104 - *GPS Reference Station Parameters* [12]

Campo	Unidade
<i>ECEF X-COORDINATE</i>	m
<i>ECEF Y-COORDINATE</i>	m
<i>ECEF Z-COORDINATE</i>	m

3.6 Conclusão

Neste capítulo foram descritas as características dos receptores utilizados e foram apresentados, de forma detalhada, todos os protocolos de comunicação que estão implementados no sistema de seguimento de plataformas móveis: o protocolo binário SiRF, o protocolo de difusão de dados SBAS RTCA DO-229, o protocolo de difusão do serviço diferencial para NAVSTAR-GPS RTCM SC-104 e o protocolo de interação com o SISNeT.

Capítulo 4

Configuração do Ambiente de Desenvolvimento

Neste capítulo descreve-se o processo de instalação e configuração do ambiente de desenvolvimento escolhido - o NetBeans Integrated Development Environment (IDE). Também são referidas outras aplicações reutilizadas na prossecução dos diversos objectivos desta dissertação.

4.1 Ambiente NetBeans IDE 6.9

Para o desenvolvimento deste trabalho foi escolhido o ambiente de desenvolvimento NetBeans IDE versão 6.9, distribuição *Java Standard Edition* (SE), que pode ser obtido em <http://netbeans.org/index.html>. Após a descarga do ficheiro executável, foram seguidos os seguintes passos para instalar o ambiente de desenvolvimento:

1. Aceitar os termos da licença de utilização do NetBeans;
2. Escolher a localização da instalação;
3. Apontar para a localização da instalação *Java Development Kit* (JDK) e do *Java Runtime Environment* (JRE) (se não estiverem instalados, é necessário fazer a descarga de <http://developers.sun.com/downloads/>);
4. Iniciar o processo de instalação.

Após estes passos, o NetBeans IDE 6.9 encontra-se instalado e é possível iniciar o desenvolvimento de aplicações, neste caso, em Java, como mostra a Figura (4.1).

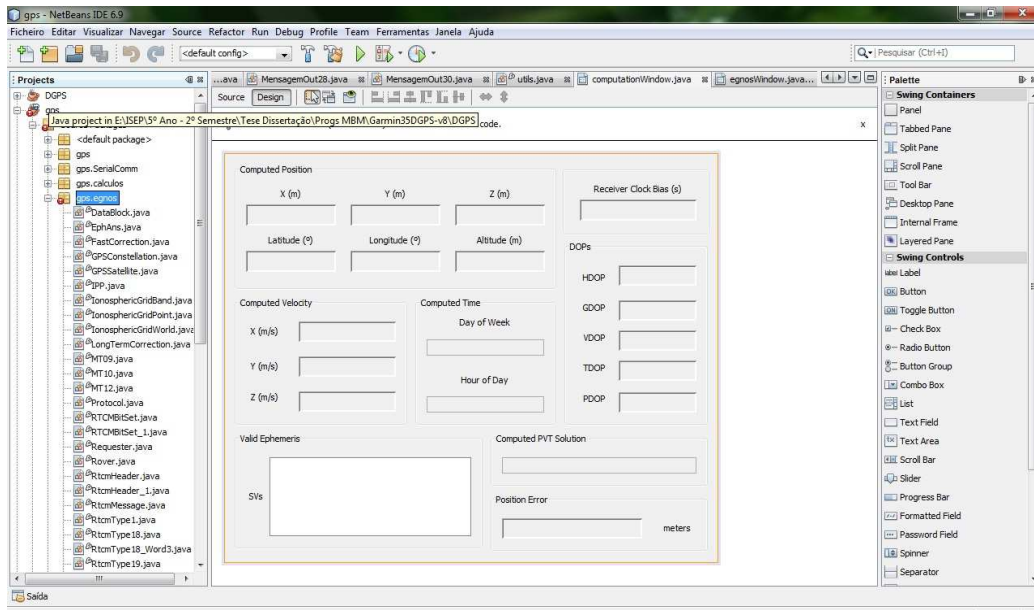


Figura 4.1: Desenvolvimento de aplicações com o NetBeans IDE

4.2 MySQL

Para assegurar o armazenamento persistente de dados escolheu-se servidor de bases de dados MySQL. Trata-se de um sistema de gestão de bases de dados relacionais muito popular. A versão instalada foi a *Community Edition*, que é *open source* e *freeware*. Possui um conjunto de ferramentas adicionais de apoio ao desenvolvimento como o MySQL Workbench e diversos controladores de acesso. No caso deste trabalho, instalou-se também a ferramenta para o desenvolvimento da base de dados em MySQL, MySQL Workbench, a API Connector J e o ODBC Connector. O MySQL Workbench é um ambiente integrado de desenvolvimento que permite:

- Conceção e modelação de bases de dados;
- Desenvolvimento de código em linguagem MySQL;
- Administração de bases de dados.

A versão *Community Edition* do MySQL Workbench instalada foi descarregada de <http://www.mysql.com/> e o processo de instalação foi feito de acordo com os seguintes passos:

1. Escolha do tipo de instalação: completa ou personalizada (foi escolhida a instalação completa);
2. Escolha da localização da instalação;
3. Início do processo de instalação.

Seguidos estes passos, é então possível iniciar a concepção, modelação e administração de bases de dados no MySQL Workbench, como ilustrado na Figura (4.2).

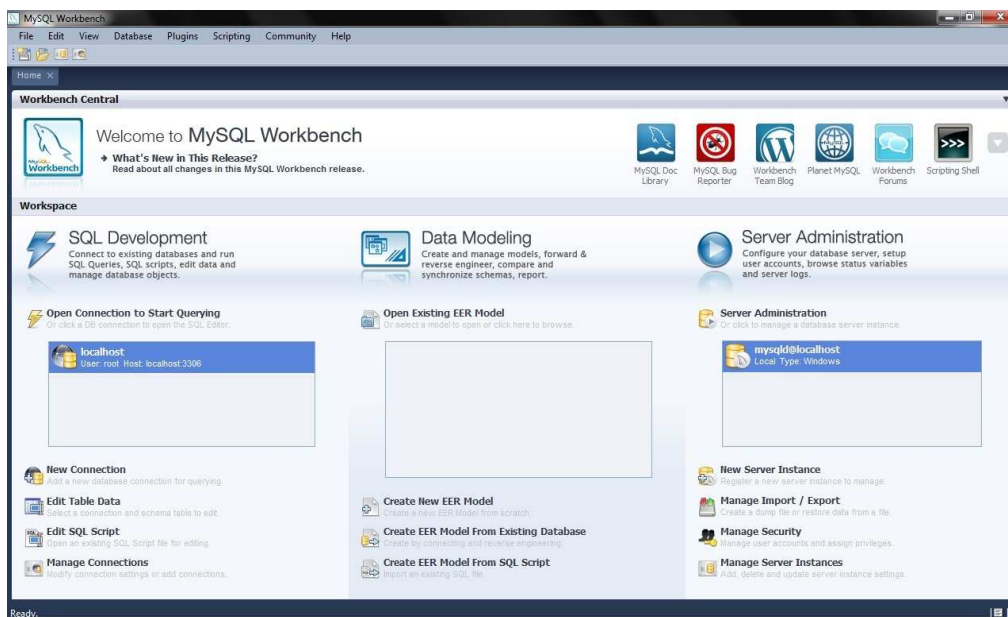


Figura 4.2: Modelação de bases de dados com o MySQL Workbench

4.3 Servidor de Aplicações Tomcat

O Apache Tomcat é um servidor de aplicações *open source* da Apache Software Foundation que implementa as tecnologias Java Servlet e Java Server Pages de desenvolvimento de aplicações *Web* servidoras. Foi escolhido como servidor da aplicação *Web* desenvolvida, não só por se tratar de uma ferramenta gratuita, como também por ser uma aplicação muito popular e com provas dadas de fiabilidade e segurança.

A instalação do Tomcat é feita através do repositório da Apache Software Foundation em <http://tomcat.apache.org/>. Pode ser instalado de três formas diferentes:

- Como um serviço: em que o Tomcat é instalado e iniciado automaticamente quando o sistema operativo é iniciado;
- Na localização do Java: em que o Tomcat utiliza os ficheiros de registo da variável ambiente `JAVA_HOME` para determinar o caminho do JRE;
- Como ícone da barra de tarefas: em que o Tomcat é controlado através desse ícone, *i.e.*, o serviço é directamente controlável pelo utilizador.

Durante o processo de instalação, o Tomcat cria todos os atalhos necessários ao controlo e configuração do servidor de aplicações. É importante notar que a administração de aplicações *Web* alojadas só pode ser feita após o Tomcat ter sido iniciado e se encontrar em execução.

4.4 Aplicação SiRF

A aplicação *Sirf* foi desenvolvida no ISEP por Jorge Rocha em 2008 sob orientação de Benedita Malheiro e visa a disponibilização, a clientes remotos e em tempo útil, dos dados provenientes de receptores GNSS que utilizam o protocolo de comunicação SiRF. Contém cinco grandes pacotes de código, cuja hierarquia é apresentada na Figura (4.3).

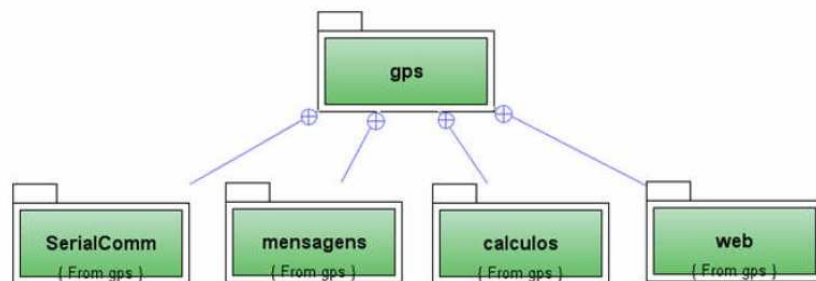


Figura 4.3: Estrutura da aplicação *Sirf*

O pacote principal `gps` reúne todas as classes necessárias para a interacção com o utilizador, nomeadamente, a classe principal `Principal` e as classes que constroem a interface gráfica com o utilizador. O pacote `SerialComm` é responsável pela interacção com o receptor GNSS através da porta série. O pacote `mensagens` contém as classes que implementam as diferentes mensagens SiRF de entrada e saída que a aplicação suporta. O pacote `calculos` possui classes complementares que efectuem os diversos cálculos necessários ao processamento das mensagens SiRF. Por último, o pacote `web` reúne as classes que permitem a interacção cliente-servidor. Esta hierarquia foi desenvolvida para suportar quatro

processos principais: (i) a aquisição, decodificação e armazenamento temporário dos dados do receptor; (ii) a solicitação de mensagens ao receptor; (iii) a interpretação gráfica; e (iv) a apresentação dos dados ao utilizador e o serviço de mensagens [15].

A aquisição e decodificação de dados é um processo fundamental da aplicação *Sirf* que é realizado através das classes dos pacotes `SerialComm` e `mensagens`. No `SerialComm` existem duas classes responsáveis pela gestão interface série: a classe que gere as portas série (`SerialComm`) e a classe que representa a ligação a uma porta (`SComm`). A classe `SComm` realiza a leitura contínua dos dados da porta série seleccionada através da escuta de eventos `SerialPortEventListener` e da interface `Runnable`. A chegada de dados à porta série é processada pelo método `serialEvent()`, que é responsável por ler da porta byte a byte, detectar a sequência inicial da mensagem, guardar os bytes num *buffer* até detectar a sequência final da mensagem e, por fim, transformar o conteúdo do *buffer* numa instância da classe da respectiva mensagem [15].

O conjunto de classes que representam as mensagens do protocolo SiRF que a aplicação suporta foram colocados no pacote `mensagens`, que possui uma classe que representa a estrutura genérica de uma mensagem SiRF, designada por `GPSSmensagem` e um conjunto de subclasses que representam cada tipo específico de mensagem. A `GPSSmensagem` define então a estrutura geral de uma mensagem SiRF, *i.e.*, dois bits de sequência inicial, o comprimento do *payload*, o *payload*, o *checksum* do *payload* e dois bits de sequência final. O *payload* é, por sua vez, transformado na instância da classe correspondente ao tipo de mensagem que contém. Nas Figuras (4.4), (4.5) e (4.6) apresenta-se a recepção de algumas mensagens no ambiente gráfico da aplicação.

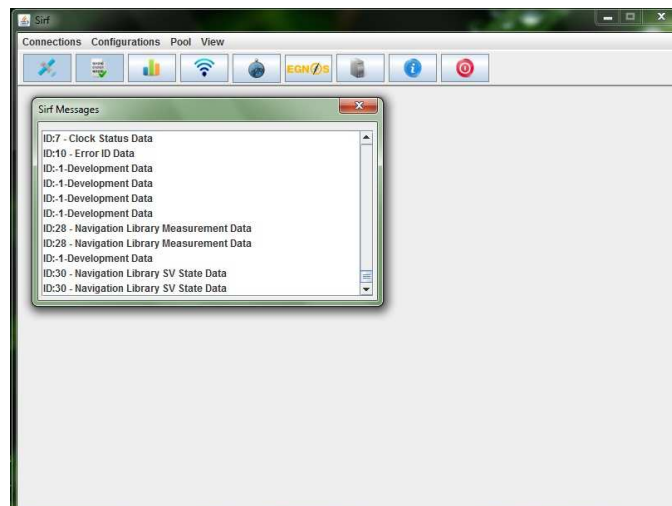


Figura 4.4: Lista das mensagens SiRF recebidas

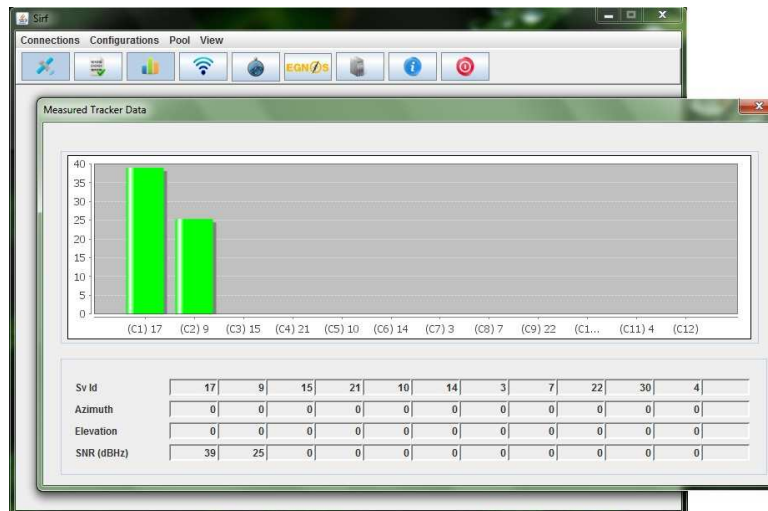


Figura 4.5: Mensagem SiRF 4 - *Measured Tracker Data*

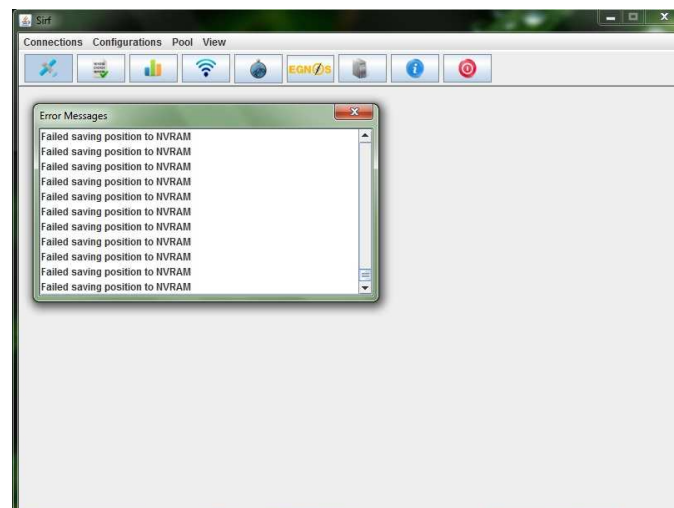


Figura 4.6: Mensagem SiRF 10 - *Error ID*

Uma das características importantes do protocolo SiRF é permitir a solicitação de mensagens. O envio de dados para o receptor é essencial para configurar o seu modo de funcionamento bem como para solicitar informação específica (como, por exemplo, as efemérides dos satélites). Todas as mensagens destinadas ao receptor utilizam o método `messageAssembler()`, que as transforma num vector de bytes correspondente do protocolo binário SiRF, e o método `enviarMensagemPoll()` que envia o vector de bytes resultante para a interface série. Estes métodos são invocados tanto por acções do utilizador, através da interface gráfica, como pelo próprio programa durante a fase do arranque da apli-

cação, quando é definida a configuração por omissão do receptor. Esta configuração por omissão é efectuada através do método `setMessageRate()` que define a identificação da mensagem e o intervalo de tempo entre mensagens consecutivas do mesmo tipo [15]. Na Figura (4.7) mostra-se o conjunto de mensagens que podem ser solicitadas e na Figura (4.8) apresenta-se uma dessas mensagens.

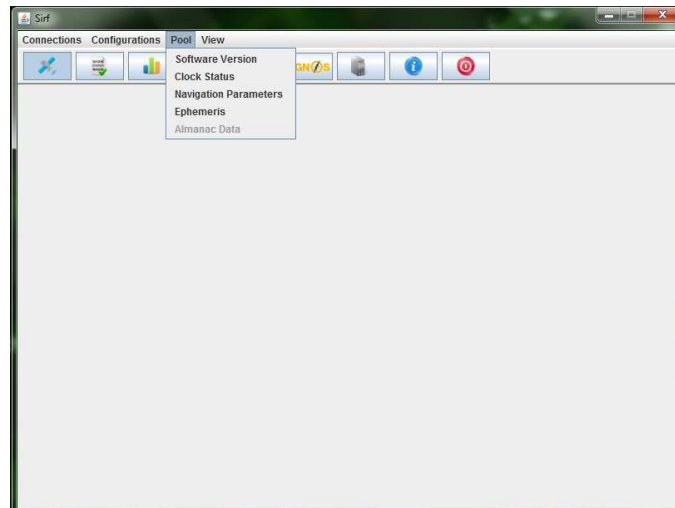


Figura 4.7: Mensagens SiRF que podem ser solicitadas

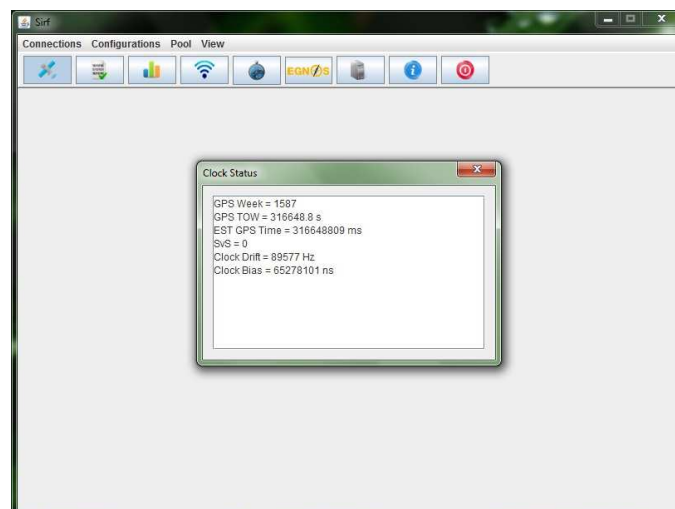


Figura 4.8: Mensagem SiRF 144 - *Poll Clock Status*

Esta aplicação foi o ponto de partida para o desenvolvimento do sistema de seguimento de plataformas móveis pois permite interagir com o receptor. Deste modo, o estudo detalhado efectuado sobre o seu funcionamento foi de elevada

importância para a prossecução dos objectivos deste trabalho.

4.5 Aplicação ReadRawData

A aplicação `ReadRawData` foi concebida para fazer a aquisição, descodificação e tratamento de dados provenientes de receptores Garmin.

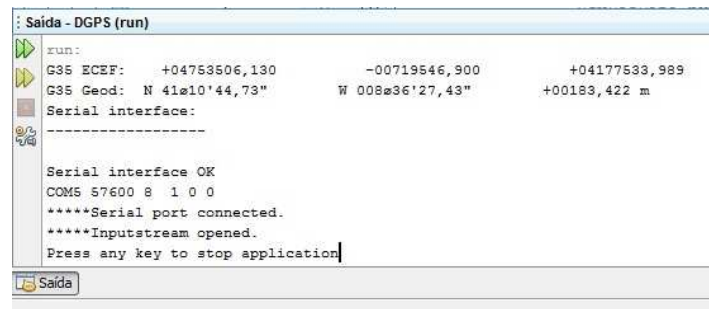
Esta aplicação, representada pela classe principal `ReadRawData`, interage com um receptor GNSS Garmin G35 através de uma interface série, representada pela classe `SerialSource`, e é capaz de obter e descodificar a solução PVT, os dados brutos e as efemérides dos satélites através das classes `RcvData`, `RcvRawData`, `RcvPosData` e `EphemerisData`. Uma vez recebidos, os dados são processados e guardados em ficheiros de registo, mediante a sua natureza: os dados brutos são guardados em ficheiros `raw`, as efemérides são guardadas em ficheiros `eph` e o posicionamento em ficheiros `pvt`.

A aplicação é também capaz de, utilizando estes dados, calcular um conjunto variado de parâmetros:

- A pseudodistância filtrada de cada satélite, técnica que consiste na filtragem das observações da pseudodistância usando as observações de fase da portadora e resulta num aumento significativo da precisão das medidas, através do método `smoothPseudoRange()`;
- A posição dos satélites visíveis com base nas suas efemérides e nas observações efectuadas através do método `calcSatPos()`;
- O azimute e elevação de cada satélite relativamente à posição do receptor em `calcAzEl()`;
- Os atrasos provados pelos desvios dos relógios dos satélites, pelas camadas da ionosfera e troposfera e a respectiva correcção a aplicar à pseudodistância observada através do método `navCorr()`;
- A posição do receptor com base na posição dos satélites visíveis através do método `solve()`.

Por fim, esta aplicação é também capaz de calcular correcções diferenciais, utilizando o método `dgpsCorr()`, gerar mensagens RTCM SC-104 com base nos resultados obtidos e aplicar as correcções às medidas observadas.

Estes algoritmos têm especial relevância para este trabalho, uma vez que a sua implementação será necessária para atingir alguns dos objectivos propostos. Por esse motivo, esta aplicação foi um exemplo a seguir. A Figura (4.9) mostra a `ReadRawData` em funcionamento, sendo evidenciada a leitura de dados do receptor através da porta série.



```
Saída - DGPS (run)
run:
G35 ECEF: +04753506,130      -00719546,900      +04177533,989
G35 Geod: N 41°10'44,73"    W 008°36'27,43"    +00183,422 m
Serial interface:
-----
Serial interface OK
COMS 57600 8 1 0 0
*****Serial port connected.
*****Inputstream opened.
Press any key to stop application
```

Figura 4.9: ReadRawData: execução

4.6 Aplicação *SISNetClient/RTCA2RTCM*

Esta aplicação consiste na fusão de dois projectos desenvolvidos no ISEP por Jelle Stuyvaert em 2006 e Tom Davidson em 2007 sob orientação de Benedita Malheiro e é capaz de obter correcções de área alargada do EGNOS e convertê-las em tempo útil do formato RTCA DO-229 para o formato RTCM SC-104.

Segundo [23], o módulo *SISNetClient* é capaz de:

- Ligar-se e autenticar-se junto do serviço *SISNet*;
- Solicitar, receber e decodificar os dados EGNOS recebidos do *SISNet* em tempo útil.

Segundo [13], o módulo *RTCA2RTCM* é capaz de:

- Processar a informação recebida do *SISNetClient*, determinando uma estimativa dos erros que se aplicam ao ponto escolhido da zona de serviço do EGNOS;
- Combinar os valores obtidos destas estimativas com os valores das mensagens RTCA DO-229 enviadas pelo *SISNet*;
- Calcular as correcções às pseudodistâncias de cada satélite com base na informação anteriormente combinada;
- Codificar as correcções para o formato RTCM SC-104 e enviá-las para a interface série.

A classe principal *SISNetClient* cria as estruturas de dados para comportar a informação recebida e estabelece a comunicação com o *SISNet* através do endereço 131.176.49.142 e do porto 7777, 7778 ou 7779. Uma vez estabelecida a ligação, lança-se a *thread* que escuta a chegada dos dados e os processa em tempo

útil através do método `processAnswer()`. Simultaneamente, é também iniciada outra *thread* intitulada `Requester` que: (i) procede, inicialmente, à autenticação junto do SISNeT - ver Figura (4.10); e (ii) solicita, periodicamente, todas as mensagens necessárias ao SISNeT.

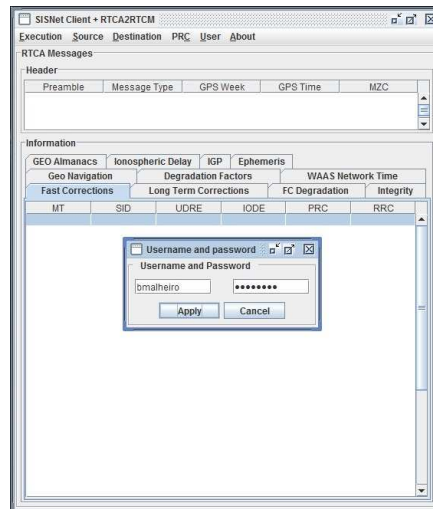


Figura 4.10: SISNetClient/RTCA2RTCM: autenticação no serviço SISNeT

A extração da informação e o seu mapeamento nas respectivas mensagens RTCA DO-229 é feito através da classe `DataBlock`, pelos métodos `getPreamble()` e `getMT()`. Dependendo do tipo de mensagem, são criadas instâncias das classes `EphAns`, `MsgAns` e `IonoAns`.

Após a extração da informação das mensagens RTCA DO-229, importa gerar as mensagens RTCM SC-104. O método `generateRTCM()` combina os dados das correções rápidas e das correções de longo prazo com as estimativas dos erros que se aplicam ao local geográfico onde se encontra o receptor. O resultado é então codificado numa mensagem RTCM SC-104, invocando o método `makeRtcmMessage()`.

Como se pode constatar, as funcionalidades implementadas nesta aplicação são essenciais para o desenvolvimento do trabalho proposto. Na Figura (4.11) apresenta-se a aplicação `SISNetClient/RTCA2RTCM` em funcionamento, evidenciando a chegada, tratamento e descodificação dos dados de área alargada do EGNOS.

4.7 Conclusão

Neste capítulo descreveu-se o ambiente de desenvolvimento adoptado e foram apresentadas as aplicações pré-existentes reutilizadas na elaboração do sistema

Capítulo 5

Desenvolvimento do Sistema

A aplicação visa o desenvolvimento de um sistema distribuído de seguimento de plataformas móveis onde é realizada: (i) a aquisição, descodificação e tratamento dos dados emitidos pelo GNSS e das observações efectuadas pelo utilizador; (ii) se aplicam as correcções de área alargada do EGNOS, recebidas através da Internet e convertidas em tempo útil do formato RTCA DO-229 para o formato RTCM SC-104. Esta última operação é efectuada segundo o conceito de Inverted Differential que realiza a aplicação das correcções diferenciais às medidas dos utilizadores num nó central, armazena os resultados obtidos numa base de dados e, por fim, disponibiliza esta informação da localização das plataformas móveis através de uma aplicação Web, permitindo assim o seu seguimento em tempo útil e de qualquer lugar com acesso à rede.

5.1 Arquitectura da Aplicação

O sistema de seguimento de plataformas móveis desenvolvido, ilustrado na Figura (5.1), consiste num conjunto de aplicações distintas que comunicam entre si segundo o modelo cliente-servidor. O módulo **SiRF Interface** deve ser instalado nas plataformas onde se ligam os receptores GNSS e, como tal, fica alojado nas plataformas móveis. Os módulos **EGNOS VRS Server**, **IDGPS Server**, a base de dados e a aplicação *Web* encontram-se alojados numa plataforma (193.136.60.90) situada no Laboratório de Comunicações por Satélite do ISEP. De notar que todos os módulos devem possuir ligação à Internet para comunicarem entre si, uma vez que utilizam canais de comunicação sobre o protocolo de transporte *Transfer Control Protocol* (TCP).

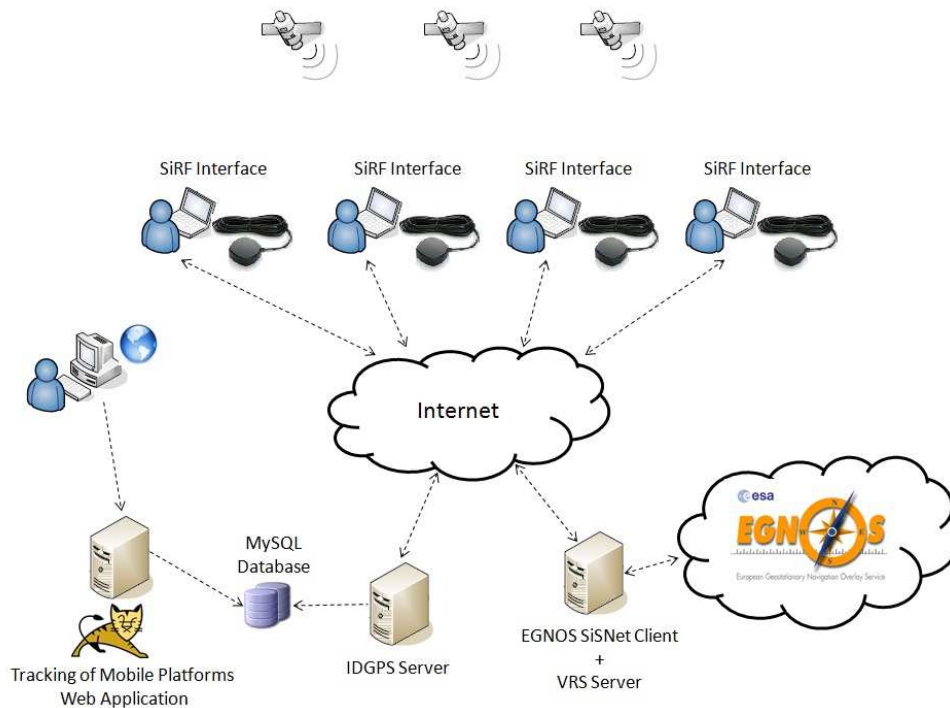


Figura 5.1: Sistema de seguimento de plataformas móveis: arquitectura

5.2 Funcionalidades da Aplicação

O trabalho desenvolvido contém quatro módulos distintos, cada um com funcionalidades específicas, e que, no seu todo, cumprem os requisitos necessários à criação de um sistema de seguimento de plataformas móveis. Na Figura (5.2) apresenta-se o diagrama funcional do sistema.

O módulo **SiRF Interface** serve de interface com os receptores GNSS e realiza a aquisição, decodificação e tratamento dos dados emitidos pelo sistema de navegação NAVSTAR-GPS.

O módulo **EGNOS VRS Server** obtém as correcções de área alargada do EGNOS através da Internet, utilizando o serviço SISNeT, no formato RTCA DO-229 e converte-as em tempo útil para o formato RTCM SC-104. Comunica com os módulos **SiRF Interface** segundo uma arquitectura cliente-servidor, recebendo a sua posição aproximada e enviando-lhes periodicamente a informação adquirida e devidamente convertida.

O módulo **IDGPS Server** é um servidor e implementa o conceito de *Inverted Differential GPS* (IDGPS), recebendo dos clientes **SiRF Interface** as observações e as respectivas correcções de área alargada do tipo VRS. Analiticamente,

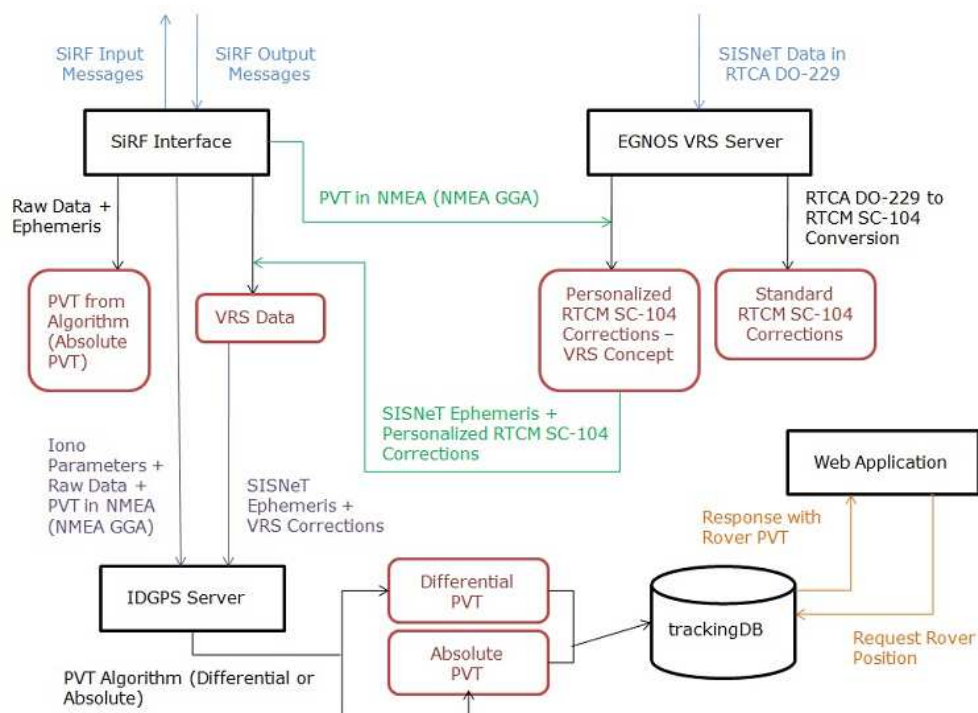


Figura 5.2: Sistema de seguimento de plataformas móveis: diagrama funcional

aplica as correções recebidas em formato RTCM SC-104 às medidas brutas enviadas pelos clientes e armazena a posição resultante numa base de dados.

A aplicação *Web* possibilita o seguimento remoto das plataformas móveis através de uma interface *Web*, ao permitir a visualização sobre um mapa da posição armazenada na base de dados de cada um dos receptores ligados.

5.2.1 Cliente SiRF Interface

Este módulo serve de interface com o receptor SiRF escolhido para este trabalho, o Holux GM-210, e realiza as seguintes funções:

1. Aquisição, decodificação e disponibilização dos dados recebidos do sistema de navegação, neste caso, o NAVSTAR-GPS;
2. Implementação das mensagens SiRF necessárias à determinação analítica da solução de posicionamento do receptor;
3. Determinação analítica da solução de posicionamento do receptor com base nos dados recebidos;

4. Solicitação e obtenção das respectivas correcções diferenciais ao módulo **EGNOS VRS Server**;
5. Submissão das correcções diferenciais, observações efectuadas e efemérides dos satélites visíveis ao módulo **IDGPS Server**.

O desenvolvimento deste módulo passou pela adaptação da aplicação desenvolvida por Jorge Rocha referida no capítulo anterior. Esta aplicação já implementava a primeira funcionalidade desejada: a aquisição, descodificação e disponibilização dos dados emitidos pelo GNSS. As restantes funcionalidades descritas nos pontos 2 a 5 foram então adicionadas à aplicação **SiRF** de modo a permitir efectuar o seguimento de plataformas móveis.

5.2.1.1 Aquisição, Descodificação e Disponibilização de Dados

Numa primeira fase, verificou-se a necessidade de depurar todo o código da aplicação **SiRF**, pois foram encontrados pequenos erros na descodificação de mensagens essenciais ao bom funcionamento deste sistema.

Os primeiros erros encontrados foram na descodificação da mensagem 28 *Navigation Library Measurement Data* que contém as medidas brutas efectuadas pelo receptor. Esta mensagem contém, para cada satélite utilizado pelo receptor, os parâmetros apresentados na Tabela (5.1):

Tabela 5.1: **SiRF MT28 - Navigation Library Measurement Data**

Campo	Unidade
<i>Channel</i>	
<i>Time Tag</i>	ms
<i>Satellite ID</i>	
<i>GPS Software Time</i>	ms
<i>Pseudomania</i>	m
<i>Carrier Frequency</i>	m/s
<i>Carrier Phase</i>	m
<i>Time in Track</i>	ms
<i>Delta Range Interval</i>	ms
<i>Mean Delta Range Time</i>	ms

Estas medidas são de vital importância pois a pseudodistância observada, o tempo de transmissão e o tempo de recepção do sinal contidos nesta mensagem são essenciais para se determinar correctamente a distância do receptor aos satélites.

Visto que os valores descodificados das observações da pseudodistância, frequência da portadora, fase da portadora e tempo de recepção não eram coerentes,

procedeu-se à aplicação das correcções prescritas em [22]. Assim, a pseudodistância observada PR - *Pseudomania* - é dada pela Equação (5.1).

Equação 5.1: Pseudodistância bruta observada pelo receptor

$$PR_{obs} = PR_{med} - CB * c \quad (m) \quad (5.1)$$

onde PR_{med} é a pseudodistância reportada pelo receptor (m), CB - *Clock Bias* - é o desvio do relógio do satélite (s) e c é a velocidade da luz no vazio dada por $2,99792458 \times 10^8$ m/s. A frequência da portadora CF é obtida através da Equação (5.2).

Equação 5.2: Frequência da portadora observada pelo receptor

$$CF_{obs} = CF_{med} - \frac{CD * c}{L1} \quad (m/s) \quad (5.2)$$

onde CF_{med} é a frequência da portadora medida pelo receptor (m/s), CD - *Clock Drift* - é a taxa de variação do desvio do relógio do satélite (s/s), c é a velocidade da luz no vazio dada por $2,99792458 \times 10^8$ m/s e $L1$ é a frequência nominal da portadora $L1 - 1575,42 \times 10^6$ Hz. A fase da portadora CP - *Carrier Phase* - é dada pela Equação (5.3).

Equação 5.3: Fase da portadora observada pelo receptor

$$CP_{obs} = CP_{med} - \frac{CB * c}{L1} \quad (m) \quad (5.3)$$

onde CP_{med} é a fase da portadora medida pelo receptor (m), CB o desvio do relógio do satélite (s) e c é a velocidade da luz no vazio dada por $2,99792458 \times 10^8$ m/s. O tempo GPS do *software* - *GPS Software Time* - observado $T_{gs_{obs}}$ é dado pela Equação (5.4).

Equação 5.4: *GPS Software Time* observado pelo receptor

$$T_{gs_{obs}} = T_{gs_{med}} - CB \quad (s) \quad (5.4)$$

onde $T_{gs_{med}}$ é o tempo GPS do *software* reportado pelo receptor e CB o desvio do relógio do satélite (s). O tempo de transmissão do satélite T_t - *Time of Transmission* - é determinado através da Equação (5.5).

Equação 5.5: Tempo de transmissão do satélite

$$T_t = T_{gs_{obs}} - \frac{PR_{obs}}{c} \quad (s) \quad (5.5)$$

onde $T_{gs_{obs}}$ é o tempo GPS do *software* da Equação (5.4), PR_{obs} é a pseudodistância observada obtida pela Equação (5.1) e c é a velocidade da luz no vácuo dada por $2,99792458 \times 10^8$ m/s. Finalmente, o tempo de recepção do receptor T_r - *Time of Reception* - é dado pela Equação (5.6).

Equação 5.6: Tempo de recepção do receptor

$$T_r = T_{gs_{obs}} \quad (s) \quad (5.6)$$

onde $T_{gs_{obs}}$ corresponde ao *GPS Software Time* observado pelo receptor e determinado pela Equação (5.4).

Estas correcções permitem que as medidas brutas não sejam degradadas devido ao desvio de relógio do receptor. De notar que os parâmetros *Clock Bias* (CB) e *Clock Drift* (CD) não são disponibilizados na mensagem 28, mas sim na mensagem 7 *Clock Status Data*. Como, segundo o protocolo SiRF, esta mensagem pode tanto ser pedida por solicitação como consecutivamente, optou-se por adicioná-la ao método `setMessageRate()` sendo assim pedida por omissão (no arranque do receptor e repetida a cada segundo). Deste modo, sempre que for recebida uma mensagem do tipo 28, os parâmetros *Clock Bias* e *Clock Drift* são actualizados e podem ser usados para a obtenção de medidas brutas correctas. O extracto de código (5.1) apresenta um excerto da classe que acomoda as mensagens do tipo 28, nomeadamente dos métodos que retornam os parâmetros corrigidos.

Excerto de Código 5.1 Excerto da classe `Mensagem28`, subclasse de `GPsmensagem`

```

1   public class MensagemOut28 extends GpsMensagem implements java.io.Serializable{
2       ...
3       public double getCarrier_Freq() {...}
4       ...
5       public double getCarrier_Phase() {...}
6       ...
7       public double getGPS_Soft_Time() {...}
8       ...
9       public double getPseudorange() {...}
10      ...
11  }
```

Seguidamente, efectuou-se uma análise da classe que acomoda as mensagens do tipo 15 *Ephemeris Data*. O correcto processamento desta mensagem é essencial para este trabalho pois são os parâmetros das efemérides dos satélites que permitem calcular as suas posições e o mais pequeno erro pode significar um desvio considerável na solução PVT final. Os parâmetros disponibilizados na mensagem 15 estão descritos na Tabela (5.2). A descodificação tem de ter em consideração as características impostas pela especificação do NAVSTAR-GPS

Tabela 5.2: SiRF MT15 - *Ephemeris Data*

Campo	Unidade
<i>Week Number - WN</i>	
<i>SV Accuracy - URA</i>	
<i>SV Health</i>	
<i>Estimated Group Delay Differential - T_{gd}</i>	s
<i>Issue of Data: Clock - IODC</i>	
<i>Time of Clock - T_{oc}</i>	s
<i>SV Clock Drift Rate - a_{f2}</i>	s/s/s
<i>SV Clock Drift - a_{f1}</i>	s/s
<i>SV Clock Bias - a_{f0}</i>	s
<i>Time of Ephemeris - T_{oe}</i>	s
<i>Eccentricity - ecc</i>	
<i>Square Root of the Semi-Major Axis - \sqrt{A}</i>	m
<i>Mean Motion Difference from Computed Value - Δn</i>	rad/s
<i>Mean Anomaly at Reference Time - M0</i>	rad
<i>Argument of Perigee - w</i>	rad
<i>Longitude of Ascending Node of Orbit Plane at Weekly Epoch - Ω_0</i>	rad
<i>Inclination Angle at Reference Time - I0</i>	rad
<i>Rate of Right Ascension - Ω_{dot}</i>	rad/s
<i>Rate of Inclination Angle - IDOT</i>	rad/s
<i>Amplitude of the Sine Harmonic Correction Term to the Argument of Latitude - C_{us}</i>	rad
<i>Amplitude of the Cosine Harmonic Correction Term to the Argument of Latitude - C_{uc}</i>	rad
<i>Amplitude of the Sine Harmonic Correction Term to the Angle of Inclination - C_{is}</i>	rad
<i>Amplitude of the Cosine Harmonic Correction Term to the Angle of Inclination - C_{ic}</i>	rad
<i>Amplitude of the Sine Harmonic Correction Term to the Orbit Radius - C_{rs}</i>	m
<i>Amplitude of the Cosine Harmonic Correction Term to the Orbit Radius - C_{rc}</i>	m
<i>Issue of Data: Ephemeris - IODE</i>	

[2]. Assim, só a correcta descodificação destes valores permite obter resultados coerentes e correctos. No extracto de código (5.2) apresenta-se um excerto respeitante à mensagem 15, evidenciando a descodificação da mensagem e algumas destas pequenas alterações realizadas.

Por fim, implementaram-se todas as mensagens que permitem comparar entre os valores calculados pelo receptor e os obtidos analiticamente através de algoritmos de posicionamento realizados neste trabalho. É o caso da mensagem 30

Excerto de Código 5.2 Excerto da classe `Mensagem15`, subclasse de `GPsmensagem`

```

1   public class MensagemOut15 extends GpsMensagem implements java.io.Serializable{
2       protected int Svid;
3       ...
4       protected float Tgd;
5       ...
6       public void inicializar(){...}
7       ...
8       public int getSvid() {
9           Svid = gps.calculos.utils.unsignedByte(Dados[2]);
10          return Svid;
11      }
12      ...
13      public float getTgd() {
14          short um = gps.calculos.utils.unsignedByte(Dados[21]);
15          //Complemento para 2
16          int value = (um & 0xff);
17          Tgd = value;
18          if(value > (Math.pow(2, 7))-1) {
19              Tgd = -(int)Math.pow(2, 8) + value;
20          }
21          Tgd = Tgd*(float)Math.pow(2, -31);
22          return Tgd;
23      }
24      ...
25  }

```

Navigation Library SV State Data que retorna os valores da posição e velocidade dos satélites num dado instante, permitindo assim a sua comparação com os resultados obtidos através do algoritmo implementado. Os dados mais relevantes que disponibiliza são apresentados na Tabela (5.3). Tal como com a mensagem

Tabela 5.3: SiRF MT30 - *Navigation Library SV State Data*

Campo	Unidade
<i>Satellite ID</i>	
<i>GPS Software Time</i>	s
<i>X Position</i>	m
<i>Y Position</i>	m
<i>Z Position</i>	m
<i>X Velocity</i>	m/s
<i>Y Velocity</i>	m/s
<i>Z Velocity</i>	m/s
<i>Ionospheric Delay</i>	m
<i>Ephemeris Flag</i>	

28 *Navigation Library Measurement Data*, também no caso desta mensagem é necessário proceder a pequenas correcções após a descodificação, nomeadamente a posição dos satélites que deve ser ajustada. A posição ajustada de um satélite é determinada através da Equação (5.7).

Equação 5.7: Posição ECEF ajustada do satélite

$$\begin{aligned} X_{adj} &= X + V_X(T_r - T_t) \quad (m) \\ Y_{adj} &= Y + V_Y(T_r - T_t) \quad (m) \\ Z_{adj} &= Z + V_Z(T_r - T_t) \quad (m) \end{aligned} \quad (5.7)$$

onde T_r é o tempo de recepção do sinal da Equação 5.4, T_t é o tempo de transmissão do sinal da Equação 5.5, X , Y , Z , V_X , V_Y e V_Z são as componentes ECEF da velocidade e da posição do satélite contidas na MT30 para o mesmo instante temporal [22]. Estas correcções previnem que a posição de cada satélite seja degradada devido à diferença provocada pelo tempo de propagação do sinal. Como esta mensagem é uma mensagem SiRF de saída, foi também adicionada ao método `setMessageRate()`, sendo assim pedida no arranque do receptor e repetida a cada segundo, garantindo assim que os instantes temporais das mensagens 28 e 30 são sempre coincidentes. As Figuras (5.3), (5.4) e (5.5) mostram o aspecto gráfico destas três mensagens, *Navigation Library Measurement Data*, *Ephemeris Data* e *Navigation Library SV State Data*, evidenciando os dados que disponibilizam e a forma como são apresentados ao utilizador.

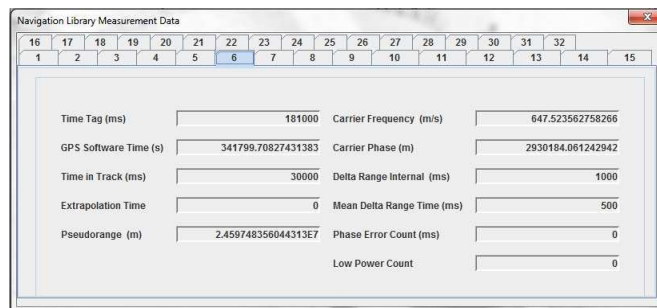


Figura 5.3: Mensagem SiRF 28 - *Navigation Library Measurement Data*

5.2.1.2 Posicionamento em Modo Absoluto

Reunidas todas as mensagens SiRF necessárias, passou-se à implementação do algoritmo de posicionamento em modo absoluto (**PosAbs**). O processo de determinação da solução PVT de um receptor apresenta as seguintes etapas:

- Determinação da posição dos satélites visíveis;
- Determinação do azimute e elevação dos satélites visíveis face ao receptor;
- Determinação das correcções a aplicar às pseudodistâncias observadas, nomeadamente correcções de relógio e de atravessamento da ionosfera e troposfera;

The screenshot shows the 'Navigation' software interface with the following data sections:

- Satellite Clock Correction Parameters:** Week Number (1595), User Range Accuracy (m) (0), SV Health (0), Iodc (70.0), Tgd (s) (-5.122274E-9), Toc (s) (345600), Clock Bias (s) (5.4645445E-4), Clock Drift (s/s) (-1.4665602E-11), Clock Drift Rate (s/s/s) (0.0).
- Position:** X (m) (2.2096875733392317E7), Latitude (°) (S 13,264183), Y (m) (1.3508704410566987E7), Longitude (°) (E 031,439114), Z (m) (-6095368.042270137), Altitude (m) (+20229573,24), Azimuth (°) (136.6109394906...), Elevation (°) (10.7509413926418...).
- Computed Correction Parameters:** Clock Delay (s) (5.465307153754908E-4), Tropospheric Delay (s) (64.65207741732424), Ionospheric Delay (s) (3.993324811504558), Corrected Pseudorange (m) (2.4672167862740304E7), Corrected Carrier Phase (m) (1793744.4404070247).
- Ephemeris Parameters:** Mean Anomaly (rad) (-1.2762076983...), Mean Anomaly Difference (rad/s) (5.453441443440017E-9), Eccentricity (0.006265088333748281), Square Root Semi-Major Axis (m*1/2) (5153.650402069092), Longitude Ascending Node (rad) (-2.983863582420076), Inclination Angle (rad) (0.9351115977343547), Argument Perigee (semi-circles) (2.1474507255538176), Rate Right Ascension (rad/s) (-8.382849179316304E-9), Rate Inclination Angle (rad/s) (3.4894310630957473E-10), Amp. Cosine Harmonic Correc. Term to Arg. Latitude (rad) (-5.327165E-7), Amp. Sine Harmonic Correc. Term to Arg. Latitude (rad) (7.8342855E-6), Amp. Cosine Harmonic Correc. Term to Orbit Radius (m) (218.78125), Amp. Sine Harmonic Correc. Term to Orbit Radius (m) (-9.125), Amp. Cosine Harmonic Correc. Term to Angle Inclination (rad) (-1.2479722E-7), Amp. Sine Harmonic Correc. Term to Angle Inclination (rad) (-6.519258E-8), Reference Time Ephemeris (s) (345600), Issue of Data (rad/s) (70.0).

Figura 5.4: Mensagem SiRF 15 - *Ephemeris Data*

- Determinação da posição do receptor com base na posição dos satélites visíveis e das pseudodistâncias corrigidas.

A posição de cada satélite numa dada época pode ser estabelecida com base nas suas efemérides. Uma vez determinado o tempo de transmissão do sinal do satélite T_t através da Equação (5.5), pode-se, de seguida, calcular a diferença T_k entre o tempo de transmissão T_t e o tempo das efemérides pela Equação (5.8).

Equação 5.8: Diferença dos tempos de transmissão e das efemérides do satélite

$$T_k = T_t - T_{oe} \quad (s) \quad (5.8)$$

onde T_{oe} é o tempo das efemérides. De notar que é necessário ter em consideração as mudanças da semana GPS nesta diferença temporal. Seguidamente, determina-se o valor do semi-eixo maior A e do movimento médio n_0 , valores dados pelas Equações (5.9) e (5.10).

Equação 5.9: Semi-eixo maior

$$A = \sqrt{A^2} \quad (5.9)$$

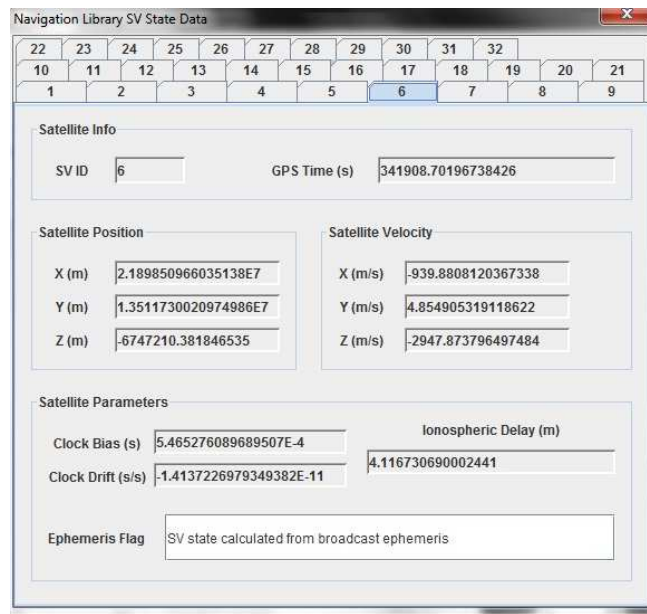


Figura 5.5: Mensagem SiRF 30 - *Navigation Library SV State Data*

Equação 5.10: Movimento médio

$$n_0 = \sqrt{\frac{\mu}{A^3}} \quad (rad/s) \quad (5.10)$$

onde μ é a constante gravitacional da Terra dada por $3,986005 \times 10^{14} \text{ m}^3/\text{s}^2$ e \sqrt{A} é um parâmetro das efemérides. Com o movimento médio calcula-se o movimento médio corrigido n e a anomalia média M_k através das Equações (5.11) e (5.12).

Equação 5.11: Movimento médio corrigido

$$n = n_0 + \Delta_n \quad (rad/s) \quad (5.11)$$

Equação 5.12: Anomalia média

$$M_k = M_0 + n * T_k \quad (rad) \quad (5.12)$$

onde Δn e M_0 são parâmetros das efemérides. A determinação da anomalia excêntrica E_k é efectuada de forma iterativa a partir de M_k e da Equação de Kepler apresentada em (5.13).

Equação 5.13: Anomalia excêntrica

$$M_k = E_k - ecc * \sin E_k \quad (rad) \quad (5.13)$$

onde ecc , a excentricidade da órbita, é um parâmetro das efemérides. Seguidamente pode calcular-se a anomalia verdadeira V_k através da Equação (5.14).

Equação 5.14: Anomalia verdadeira

$$V_k = \arctan \frac{\sin V_k}{\cos V_k} = \arctan \frac{\frac{\sqrt{1-ecc^2} * \sin E_k}{1-ecc * \cos E_k}}{\frac{\cos E_k - ecc}{1-ecc * \cos E_k}} \quad (5.14)$$

Com V_k é possível calcular o argumento da latitude ϕ_k pela Equação (5.15).

Equação 5.15: Argumento da latitude

$$\phi_k = V_k + w \quad (5.15)$$

onde w , designado argumento do perigeu, é um parâmetro das efemérides. O argumento da latitude ϕ_k é então usado para calcular diversos parâmetros como o argumento da correcção da latitude δ_{U_k} , a correcção do raio δ_{R_k} e a correcção da inclinação δ_{I_k} , através das Equações (5.16), (5.17) e (5.18).

Equação 5.16: Argumento da correcção da latitude

$$\delta_{U_k} = C_{us} * \sin 2 * \phi_k + C_{uc} * \cos 2 * \phi_k \quad (5.16)$$

Equação 5.17: Argumento da correcção do raio

$$\delta_{R_k} = C_{rs} * \sin 2 * \phi_k + C_{rc} * \cos 2 * \phi_k \quad (5.17)$$

Equação 5.18: Argumento da correcção da inclinação

$$\delta_{I_k} = C_{is} * \sin 2 * \phi_k + C_{ic} * \cos 2 * \phi_k \quad (5.18)$$

onde C_{us} , C_{rs} , C_{is} , C_{uc} , C_{rc} e C_{ic} são os termos de correcção dos senos e cossenos da latitude, raio e inclinação dados pelas efemérides. Com estes valores é possível calcular os valores corrigidos do argumento da latitude U_k , do raio R_k e da inclinação I_k através das Equações (5.19), (5.20) e (5.21).

Equação 5.19: Latitude corrigida

$$U_k = \phi_k + \delta_{U_k} \quad (5.19)$$

Equação 5.20: Raio corrigido

$$R_k = A * (1 - ecc * \cos E_k) + \delta_{R_k} \quad (5.20)$$

Equação 5.21: Inclinação corrigida

$$I_k = i_0 + IDOT * T_k + \delta_{I_k} \quad (5.21)$$

onde i_0 e $IDOT$, que são parâmetros do ângulo de inclinação, são dados das efemérides. Assim, é possível determinar as coordenadas orbitais X_k e Y_k do satélite através da Equação (5.22).

Equação 5.22: Coordenadas orbitais X e Y do satélite

$$\begin{aligned} X_k &= R_k * \cos U_k \\ Y_k &= R_k * \sin U_k \end{aligned} \quad (5.22)$$

Com as coordenadas orbitais, é possível determinar a longitude corrigida do nodo ascendente Ω_k pela Equação (5.23).

Equação 5.23: Longitude corrigida do nodo ascendente

$$\Omega_k = \Omega_0 + (\Omega_{dot} - \Omega_{edot}) * T_k + \Omega_e * T_{oe} \quad (5.23)$$

onde Ω_0 e Ω_{dot} são parâmetros das efemérides e Ω_{edot} é a velocidade de rotação da Terra dada por $7,2921151467 \times 10^{-5}$ rad/s. Finalmente, as coordenadas ECEF do satélite são obtidas através da Equação (5.24).

Equação 5.24: Coordenadas ECEF do satélite

$$\begin{aligned} X &= X_k * \cos \Omega_k - Y_k * \sin \Omega_k * \cos I_k \\ Y &= X_k * \sin \Omega_k + Y_k * \cos \Omega_k * \cos I_k \\ Z &= Y_k * \sin \Omega_k \end{aligned} \quad (5.24)$$

Estas equações foram implementadas no método `inicializaXYZ()` da classe `Mensagem15` que recebe como parâmetros de entrada as efemérides do satélite em

questão. Os valores retornados são armazenados através dos métodos `setX()`, `setY()` e `setZ()` desta classe. De notar que as expressões utilizadas neste algoritmo se encontram definidas na especificação do NAVSTAR-GPS [2].

Calculadas as coordenadas ECEF X , Y , Z de cada satélite, é necessário determinar o respectivo azimute e elevação em relação ao receptor. Para tal, muda-se do sistema de coordenadas ECEF para um sistema *East-North-Up* (ENU) com origem no receptor. Deste modo, obtém-se a posição do satélite relativamente ao receptor como se ilustra na Figura (5.6). Isto é realizado através do método `inicializaAzEl()` da classe `Mensagem15` que recebe como parâmetros de entrada as coordenadas ECEF do satélite e do receptor. Os valores retornados, em radianos, são armazenados através dos métodos `setAz()` e `setEl()` desta classe.

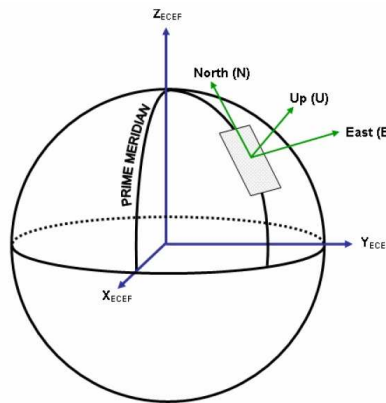


Figura 5.6: Representação de coordenadas ECEF e ENU

A pseudodistância bruta observada pelo receptor contém erros provocados pelo desvio do relógios do satélite e pelo atravessamento das camadas atmosféricas que podem ser estimados e retirados à medida efectuada. O desvio do relógio do satélite dT_{clk} é obtido através das Equações (5.25), (5.26) e (5.27) [2].

Equação 5.25: Diferença entre o tempo de transmissão e do relógio do satélite

$$T = T_t - T_{oc} \quad (s) \quad (5.25)$$

onde T é o intervalo entre o tempo de transmissão T_t da Equação (5.5) e o tempo de relógio do satélite T_{oc} contido nas efemérides.

Equação 5.26: Termo de correcção relativista

$$T_{rel} = F * ecc * \sqrt{A} * \sin E_k \quad (s) \quad (5.26)$$

onde T_{rel} , o termo de correcção relativista, é determinado a partir dos parâmetros das efemérides ecc e A e da anomalia excêntrica E_k da Equação (5.14).

Equação 5.27: Desvio de relógio do satélite

$$dT_{clk} = a_{f0} + a_{f1} * T + a_{f2} * T^2 + T_{rel} - T_{gd} \quad (s) \quad (5.27)$$

onde T é o intervalo de tempo dado pela Equação (5.25), T_{rel} é o termo de correcção relativista da Equação (5.26) e a_{f0} , a_{f1} , a_{f2} e T_{gd} são parâmetros do relógio do satélite contidos nas efemérides. O desvio provocado pelo atravessamento da ionosfera dT_{iono} deve ser determinado através dos parâmetros do modelo ionosférico [2]. O factor de obliquidade F_{ob} e o ângulo central da Terra ψ entre a posição do receptor e a projecção terrestre do ponto de intersecção da ionosfera - *Ionospheric Pierce Point* (IPP) - são obtidos pelas Equações (5.28) e (5.29).

Equação 5.28: Factor de obliquidade da ionosfera

$$F_{ob} = 1 + 16 * (0,53 - \frac{Sat_{el}}{\pi})^3 \quad (5.28)$$

Equação 5.29: Ângulo central do receptor com a projecção terrestre do IPP

$$\psi = \frac{0,0137}{\frac{Sat_{el}}{\pi} - 0,11} - 0,222 \quad (semi-círculos) \quad (5.29)$$

onde Sat_{el} é o valor da elevação do satélite em radianos. Os valores da latitude e longitude geodésica da projecção terrestre do ponto de intersecção da ionosfera ϕ_i e λ_i , respectivamente, são obtidos através das Equações (5.30) e (5.31).

Equação 5.30: Latitude geodésica da projecção terrestre do IPP

$$\phi_i = \phi_{WGS84} + \psi * \cos \frac{Sat_{az}}{\pi} \quad (semi-círculos) \quad (5.30)$$

Equação 5.31: Longitude geodésica da projecção terrestre do IPP

$$\lambda_i = \lambda_{WGS84} + \frac{\psi * \sin \frac{Sat_{az}}{\pi}}{\cos \phi_i} \quad (semi-círculos) \quad (5.31)$$

onde ϕ_{WGS84} e λ_{WGS84} são a latitude e longitude no *World Geodetic System* 1984 (WGS 84) e Sat_{az} é o valor do azimute do satélite em radianos. Com estes

parâmetros é possível determinar a latitude geomagnética da projecção terrestre do ponto de intersecção da ionosfera ϕ_m e o tempo local t através das Equações (5.32) e (5.33).

Equação 5.32: Latitude geomagnética da projecção terrestre do IPP

$$\phi_m = \phi_i + 0,064 * \cos \phi_i - 1,617 \quad (\text{semi-círculos}) \quad (5.32)$$

Equação 5.33: Tempo local

$$t = 4,32 * 10^4 * \lambda_i + T_t \quad (s) \quad (5.33)$$

em que T_t é tempo de transmissão do sinal obtido através da Equação (5.5). Os parâmetros ionosféricos PER , AMP e x são obtidos através das Equações (5.34), (5.35) e (5.36).

Equação 5.34: Parâmetro da ionosfera PER

$$PER = \beta_0 + \beta_1 * \phi_m + \beta_2 * \phi_m^2 + \beta_3 * \phi_m^3, PER \geq 7200 \quad (5.34)$$

Equação 5.35: Parâmetro da ionosfera AMP

$$AMP = \alpha_0 + \alpha_1 * \phi_m + \alpha_2 * \phi_m^2 + \alpha_3 * \phi_m^3, AMP \geq 0 \quad (5.35)$$

Equação 5.36: Parâmetro da ionosfera x

$$x = \frac{2 * \pi * (t - 50400)}{PER} \quad (5.36)$$

onde α_n e β_n são os parâmetros ionosféricos, que no caso desta aplicação, são obtidos a partir dos ficheiros RINEX disponibilizados em <http://navstar.idt.ipp.pt/> e gerados pela estação de referência do Instituto de Desenvolvimento Tecnológico (IDT) localizada no *campus* do ISEP. Desta forma, o atraso provocado pela ionosfera dT_{iono} é obtido pela Equação (5.37).

Equação 5.37: Atraso ionosférico

$$\begin{aligned} dT_{iono} &= F_{ob} * (5,0 * 10^{-9} + AMP * (1 - \frac{x^2}{2} + \frac{x^4}{24})) \quad (s), |x| < 1,57 \\ dT_{iono} &= F_{ob} * 5,0 * 10^{-9} \quad (s), |x| \geq 1,57 \end{aligned} \quad (5.37)$$

Por último, o atraso provocado pelo atravessamento da troposfera é calculado com base num algoritmo que tem em conta as componentes húmida e seca da atmosfera retirado da aplicação `ReadRawData`. Assim, o atraso troposférico dRT_{trop} é dado pelo modelo de Hopfield [24] para parâmetros padrão da atmosfera ($T_{amb} = 15^\circ\text{C}$, $P_{amb} = 101,32\text{ kPa}$ e $P_{vap} = 0,85\text{ kPa}$) segundo a Equação (5.38).

Equação 5.38: Atraso troposférico

$$dRT_{trop} = \frac{2,312}{\sin Sat_{el}^2 + 1,904^{-3}} + \frac{0,084}{\sin Sat_{el}^2 + 0,6854^{-3}} \quad (m) \quad (5.38)$$

em que Sat_{el} é a elevação do satélite em radianos. Assim, a pseudodistância corrigida dos erros do relógio do satélite e do atravessamento das camadas da atmosfera PRC_{abs} pode ser calculada pela Equação (5.39).

Equação 5.39: Pseudodistância corrigida em modo absoluto

$$PRC_{abs} = PR_{obs} + dT_{clk} * c - dT_{iono} * c - dRT_{trop} \quad (m) \quad (5.39)$$

em que PR_{obs} é a pseudodistância observada determinada pela Equação (5.1) e c é a velocidade da luz no vácuo - $2,99792458 \times 10^8\text{ m/s}$. A PRC_{abs} é a pseudodistância utilizado no cálculo do posicionamento em modo absoluto.

Este conjunto de equações é executado no método `inicializaCorr()` da classe `Mensagem15` que recebe como parâmetros de entrada as efemérides do satélite em questão e as coordenadas ECEF do receptor. A pseudodistância corrigida resultante é armazenada através do método `setPrc()`. De notar que, tal como para o cálculo da posição de cada satélite, as expressões utilizadas neste algoritmo também se encontram definidas na especificação de interface do NAVSTAR-GPS [2].

Deste modo, estão neste momento reunidas as condições para o cálculo da posição do receptor. O algoritmo escolhido foi retirado da aplicação `ReadRawData` e utiliza uma aproximação por Série de Taylor para determinar a solução de posicionamento [25]. Para determinar a posição de um receptor GNSS, é necessário satisfazer duas condições:

1. Existirem pelo menos quatro satélites visíveis por esse receptor;
2. A elevação desses satélites ser maior do que 5° , condição estipulada pela especificação do NAVSTAR-GPS para minimizar erros.

Verificadas estas condições, o algoritmo segue a seguinte lógica:

1. Inicia a posição do receptor no centro da Terra, *i.e.*, as coordenadas X , Y e Z são iniciadas a 0, e calcula a distância do receptor a cada satélite através da Equação (5.40).

Equação 5.40: Algoritmo PosAbs: Distância do receptor ao satélite

$$R_i = \sqrt{(X_r - X_{si})^2 + (Y_r - Y_{si})^2 + (Z_r - Z_{si})^2} \quad (m) \quad (5.40)$$

em que X_r , Y_r e Z_r são as coordenadas ECEF do receptor e X_{si} , Y_{si} e Z_{si} são as coordenadas ECEF do satélite i ;

2. Determina o valor residual entre esta distância e a pseudodistância corrigida do satélite em questão;
3. Calcula, para cada direcção X , Y e Z , a respectiva componente de variação da distância entre o receptor e cada satélite através da Equação (5.41).

Equação 5.41: Algoritmo PosAbs: Componentes da variação da distância

$$\begin{aligned} A_{i0} &= \frac{X_r - X_{si}}{R_i} \\ A_{i1} &= \frac{Y_r - Y_{si}}{R_i} \\ A_{i2} &= \frac{Z_r - Z_{si}}{R_i} \end{aligned} \quad (5.41)$$

onde X_r , Y_r e Z_r são as coordenadas ECEF do receptor, X_{si} , Y_{si} e Z_{si} são as coordenadas ECEF do satélite i e R_i é a distância calculada do receptor ao satélite i ;

4. Calcula, utilizando aproximações por Série de Taylor, o DON e o desvio de posicionamento que é, então, adicionado às coordenadas X_r , Y_r e Z_r do receptor;
5. Com os novos valores X_r , Y_r e Z_r , repetem-se os pontos 2, 3 e 4 até que o erro obtido pelo desvio de posicionamento seja menor do que $1,0 * 10^{-4}$;
6. Findo este processo iterativo, os valores das coordenadas X_r , Y_r e Z_r do receptor constituem a solução da posição calculada analiticamente.

Foi adicionada à classe `Principal` uma janela designada `computationWindow` que apresenta os resultados deste cálculo. O método que executa o cálculo denomina-se `inicializaPVT()` e recebe como parâmetros de entrada as efemérides dos satélites que cumprem as condições estipuladas. No excerto de código (5.3) apresenta-se um extracto do método `inicializaPVT()`, onde é definido o algoritmo de cálculo da posição do receptor. A Figura (5.7) apresenta a janela `computationWindow` com os resultados do algoritmo `PosAbs`.

Excerto de Código 5.3 Excerto do método inicializaPVT()

```

1  public static Vector inicializaPVT(Vector eph_data) {
2  ...
3  int it = 0;
4  do {
5  it ++;
6  for(int i = 0; i < eph_data.size(); i++) {
7  gps.mensagens.MensagemOut15 dados = (MensagemOut15) eph_data.elementAt(i);
8  xs[i] = dados.getX();
9  ys[i] = dados.getY();
10 zs[i] = dados.getZ();
11 R[i] = java.lang.Math.sqrt(java.lang.Math.pow(xr - xs[i], 2)
12 + java.lang.Math.pow(yr - ys[i], 2) + java.lang.Math.pow(zr - zs[i], 2));
13 L[i] = dados.getPrc() - R[i];
14 A[i][0] = (xr - xs[i])/R[i];
15 A[i][1] = (yr - ys[i])/R[i];
16 A[i][2] = (zr - zs[i])/R[i];
17 A[i][3] = -1.0;
18 }
19 for (int j = 0; j <= 3; j++) {
20 AL[j] = 0.0;
21 for (int i = 0; i < eph_data.size(); i++) {
22 AL[j] += A[i][j] * L[i];
23 }
24 }
25 for (int j = 0; j <= 3; j++) {
26 for (int w = 0; w <= 3; w++) {
27 AA[j][w] = 0.0;
28 for (int i = 0; i < eph_data.size(); i++) {
29 AA[j][w] += A[i][j] * A[i][w];
30 }
31 }
32 }
33 det = + AA[0][0] * sub(AA,0,0) - AA[1][0] * sub(AA,1,0)
34 + AA[2][0] * sub(AA,2,0) - AA[3][0] * sub(AA,3,0);
35 if (det == 0.0) {
36 gps.Principal.compdata_log.println("Cannot invert matriz!\n");
37 gps.computationWindow.calculadoPVT = false;
38 xr = 0;
39 yr = 0;
40 zr = 0;
41 }
42 int j;
43 int n;
44 for (int k = 0; k <= 3; k++) {
45 for (int w = 0; w <= 3; w++) {
46 n = k + w;
47 if ( n % 2 != 0 )
48 j = -1;
49 else
50 j = 1;
51 AAinv[k][w] = j * sub(AA, w, k) / det;
52 }
53 }
54 for (int k = 0; k <= 3; k++) {
55 D[k] = 0.0;
56 for (int w = 0; w <= 3; w++) {
57 D[k] += AAinv[k][w] * AL[w];
58 }
59 }
60 xr = xr + D[0];
61 yr = yr + D[1];
62 zr = zr + D[2];
63 }
64 while((it < 6) && ((java.lang.Math.abs(D[0]) + java.lang.Math.abs(D[1]) +
65 java.lang.Math.abs(D[2])) >= 1.0E-4));

```

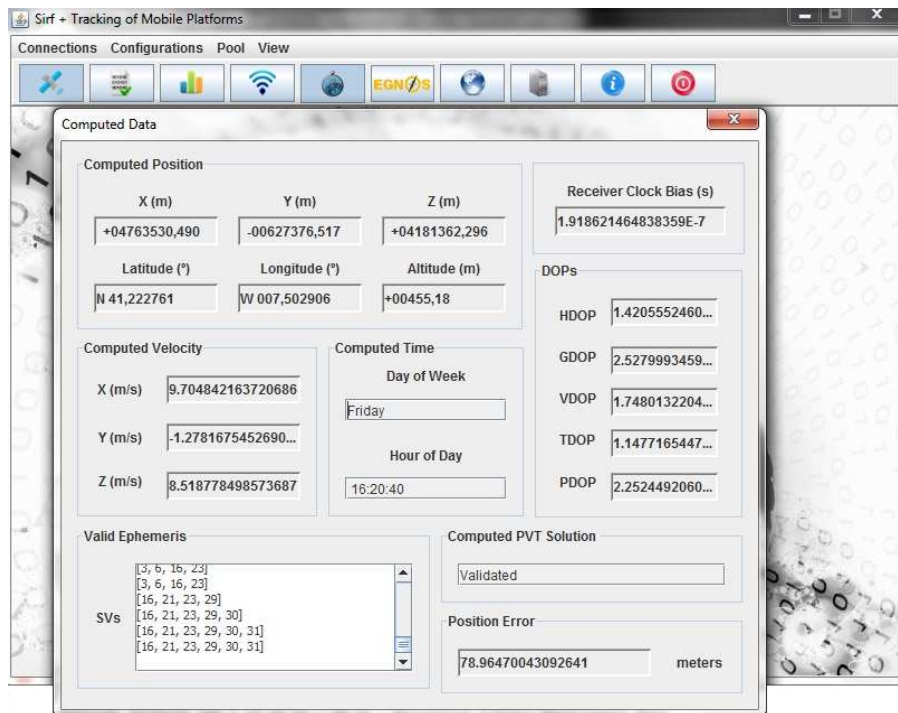


Figura 5.7: SiRF Interface: resultados do algoritmo PosAbs

5.2.1.3 Interação com o Serviço EGNOS VRS

Para evitar que cada módulo SiRF Interface solicite directamente as correcções ao SISNeT, resultando em múltiplas ligações, criou-se um novo módulo servidor para esta função: o EGNOS VRS Server. Este serviço liga-se ao SISNeT, solicita as correcções do EGNOS e utiliza-as para determinar as correcções diferenciais de cada receptor.

A ligação cliente é feita pela classe `connectEGNOS` que é executada através da classe principal do SiRF Interface. Quando iniciada, esta classe estabelece uma ligação TCP com o porto 5050 do endereço 193.136.60.90 (onde está alojado EGNOS VRS Server) e cria os respectivos canais de entrada e saída de dados. Seguidamente, são iniciadas as *threads* que vão lidar com o envio e recepção das mensagens, `SendData()` e `ReceiveData()`, respectivamente. Por fim, é enviado o pedido de identificação ao servidor, o que permite a identificação única e inequívoca de cada cliente. O extracto de código (5.4) mostra este processo.

A classe `SendData` é responsável pelo envio de dados do cliente para o servidor. Trata-se de uma *thread* que envia periodicamente as coordenadas do receptor GNSS ligado ao cliente SiRF Interface através do método `sendCoords()`. Este método extrai as coordenadas do receptor, recebidas na mensagem SiRF 4 *Mea-*

Excerto de Código 5.4 Método construtor da classe `connectEGNOS`

```

1  public class connectEGNOS extends javax.swing.JDialog {
2      private int port = 5050;
3      private String addr = "193.136.60.90";
4      ...
5      public connectEGNOS(final Principal parent) {
6          ...
7          try {
8              this.conn = new Socket(InetAddress.getByName(addr), this.port);
9              this.out_data = new ObjectOutputStream(this.conn.getOutputStream());
10             this.in_data = new ObjectInputStream(this.conn.getInputStream());
11             send = new SendData(this);
12             send.start();
13             receive = new ReceiveData(this);
14             receive.start();
15             MsgIdt msg = new MsgIdt();
16             msg.send(out_data);
17             this.display(msg);
18         }
19         catch (Exception e) {
20             e.printStackTrace();
21         }
22     }
23 }

```

sured Tracker Data, e converte-as para o formato de dados normalizado NMEA, criando uma mensagem NMEA GPGGA. Esta mensagem NMEA, contendo as coordenadas geodésicas do receptor, é então enviada para o servidor. De notar que esta conversão de SiRF para NMEA é das mais importantes características do sistema de seguimento de plataformas móveis porque o NMEA é um formato de dados normalizado e amplamente utilizado por um variado conjunto de equipamentos, conferindo assim interoperabilidade ao sistema. O extracto de código (5.5) mostra como se processa o envio de dados do cliente para o servidor através da classe `SendData`.

A recepção de dados provenientes do `EGNOS VRS Server` é feita pela classe `ReceiveData`, que também é uma *thread*. Esta classe aguarda a chegada de mensagens através do canal de entrada de dados. O método `processMsg()` recebe e encaminha devidamente as mensagens do protocolo de comunicação definido entre o servidor e cliente. Este protocolo definido na classe `ProtocolEGNOS` especifica as mensagens entre os módulos `SiRF Interface` (pedidos) e `EGNOS VRS Server` (respostas). Em particular, o `SiRF Interface` recebe:

- A resposta ao pedido de identificação enviado no método construtor que é processada pelo método `processMsgIdt()`;
- As efemérides da constelação, processadas pelo método `processMsgEph()`;
- As correcções de área alargada em formato RTCM SC-104 que são processadas pelo método `processMsgRTCM()`;
- A resposta ao pedido de fim de ligação que é processada pelo método `processMsgEnd()`.

Excerto de Código 5.5 *Thread SendData*, responsável pelo envio de dados entre o SiRF Interface e o EGNOS VRS Server

```

1  class SendData extends Thread {
2      ...
3      public SendData(connectEGNOS parent) {
4          ...
5      }
6      public void run() {
7          try {
8              while(!parent.done) {
9                  Thread.sleep(1000);
10                 sendCoords();
11             }
12         }
13         catch (Exception ex) {
14             ex.printStackTrace();
15         }
16     }
17     public void sendCoords() {
18         double xPos = gps.mensagens.MensagemOut2.getXpos();
19         double yPos = gps.mensagens.MensagemOut2.getYpos();
20         double zPos = gps.mensagens.MensagemOut2.getZpos();
21         String lat = gps.mensagens.MensagemOut2.getLat();
22         String unitLat = (gps.mensagens.MensagemOut2.getLat()).substring(0, 1);
23         String lon = gps.mensagens.MensagemOut2.getLon();
24         String unitLon = (gps.mensagens.MensagemOut2.getLon()).substring(0, 1);
25         String alt = gps.mensagens.MensagemOut2.getAlt();
26         this.parent.coordData = "GPGGA,0,"+(lat.substring(2)).replace(',','.')
27             +","+unitLat+","+((lon.substring(2)).replace(',','.'))+","+unitLon
28             +",0,0,0,"+(alt.replace(',','.'))+",M,0,M,0,0*0";
29         MsgPos msg = new MsgPos(parent.myClientID, parent.coordData);
30         try {
31             msg.send(parent.out_data);
32         }
33         catch (Exception e) {
34             e.printStackTrace();
35         }
36     }
37 }

```

Este processo encontra-se descrito no extracto de código (5.6) dedicado à recepção e processamento das mensagens recebidas do EGNOS VRS Server.

Os dados transaccionados entre o SiRF Interface e o EGNOS VRS Server são apresentadas na janela definida na classe `connectEGNOS` e ilustrada na Figura (5.8). De notar que, enquanto o fecho desta janela fecha a ligação TCP estabelecida entre os dois módulos, a sua minimização apenas oculta a janela, não interferindo com o processo de comunicação. Esta abordagem foi adoptada para facilitar a visualização das múltiplas janelas disponibilizadas pela aplicação SiRF Interface.

5.2.1.4 Interacção com o Serviço IDGPS

Nesta abordagem os receptores enviam as suas medidas brutas para um nó central onde são aplicadas as correcções diferenciais e determinadas as soluções PVT corrigidas [26]. É classificado de “invertido” face ao conceito tradicional DGNSS porque, ao invés de determinar e enviar para os receptores as correcções diferenciais, é o nó central quem recebe as medidas brutas dos receptores e lhes aplica as respectivas correcções diferenciais. Como, neste caso, o sistema de navegação uti-

Excerto de Código 5.6 *Thread ReceiveData*, responsável pela recepção de dados provenientes do EGNOS VRS Server

```

1  class ReceiveData extends Thread {
2      ...
3      public ReceiveData(connectEGNOS parent) {
4          ...
5      }
6      public void run() {
7          try {
8              while(!parent.done) {
9                  processMsg(ProtocolEGNOS.receive(parent.in_data));
10             }
11         }
12         catch (Exception e) {
13             e.printStackTrace();
14         }
15         ...
16     }
17     public void processMsg(ProtocolEGNOS msg) throws Exception {
18         if(msg instanceof MsgIdt)
19             processMsgIdt((MsgIdt)msg);
20         else if(msg instanceof MsgEph)
21             processMsgEph((MsgEph)msg);
22         else if(msg instanceof MsgRTCM)
23             processMsgRTCM((MsgRTCM)msg);
24         else if(msg instanceof MsgEnd)
25             processMsgEnd((MsgEnd)msg);
26     }
27     ...
28 }

```

lizado é o NAVSTAR-GPS, trata-se da interacção com um *Inverted Differential* GPS (IDGPS).

Neste contexto, os diversos módulos **SiRF Interface** constituem-se clientes do módulo **IDGPS Server** e são responsáveis por lhe submeter periodicamente as suas medidas brutas e as respectivas correcções diferenciais.

A ligação da aplicação **SiRF Interface** ao módulo **IDGPS Server** é estabelecida através da classe **connectIDGPS**. Esta classe cria um *socket* TCP e os respectivos canais de entrada e saída de dados com o porto 5060 do endereço 193.136.60.90 onde se encontra o **IDGPS Server** à escuta. Abertos os canais de comunicação, são iniciadas as *threads* **SendData()** e **ReceiveData()**, responsáveis pelo envio e recepção de dados, respectivamente. Este processo encontra-se representado no extracto de código (5.7).

O envio de dados do cliente **SiRF Interface** para o servidor **IDGPS Server** é mais relevante do que a recepção de dados porque, tal como foi referido, a implementação do conceito IDGPS obriga a que os receptores enviem os seus dados para um nó central que é responsável pelas restantes operações. Assim, quando um cliente **SiRF Interface** se liga ao **IDGPS Server**, segue a seguinte lógica:

- Envia o pedido de identificação ao servidor;
- Escuta a chegada da confirmação da identificação;

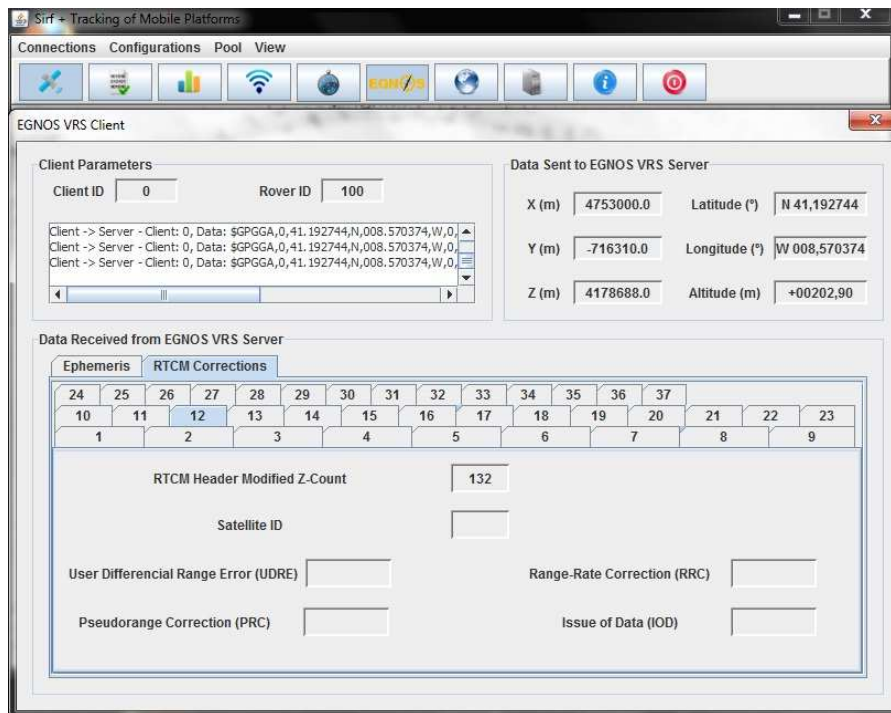


Figura 5.8: SiRF Interface: interação com o EGNOS VRS Server

- Envia os parâmetros da ionosfera;
- Envia periodicamente as coordenadas, medidas brutas, efemérides e correcções diferenciais;
- Envia o pedido de fecho de ligação;
- Escuta a indicação de fecho da ligação.

O envio de mensagens é realizado pela *thread* `SendData`. Esta classe envia a mensagem de identificação quando o cliente se liga (representada pelo método `sendIdentification()`) e, posteriormente, as mensagens com os dados das coordenadas, medidas brutas, parâmetros ionosféricos, efemérides e correcções diferenciais (representadas respectivamente pelos métodos `sendCoords()`, `sendRaw()`, `sendIono()`, `sendEph()` e `sendRTCM()`). De notar que, com a excepção das efemérides e dos parâmetros ionosféricos, as restantes mensagens são enviadas a uma frequência de 1 Hz. As efemérides são enviadas a cada 5 min e os parâmetros ionosféricos são apenas enviados quando o cliente se liga. O extracto de código (5.8) apresenta a classe `SendData` e, em particular, o método `run()` que envia as diversas mensagens.

Excerto de Código 5.7 Método construtor da classe `connectIDGPS`

```

1  public class connectIDGPS extends javax.swing.JDialog {
2      private int port = 5060;
3      private String addr = "193.136.60.90";
4      ...
5      public connectIDGPS(final Principal parent) {
6          ...
7          try {
8              this.conn = new Socket(InetAddress.getByName(addr), this.port);
9              this.out_data = new ObjectOutputStream(this.conn.getOutputStream());
10             this.in_data = new ObjectInputStream(this.conn.getInputStream());
11             send = new SendData(this);
12             send.start();
13             receive = new ReceiveData(this);
14             receive.start();
15             this.display("Connected to IDGPS Server in "
16                 +InetAddress.getByName(addr)+"", on port " + this.port + "!");
17         }
18         catch (Exception e) {
19             this.display("Problems connecting to IDGPS Server!");
20             e.printStackTrace();
21         }
22     }
23     ...
24 }

```

Excerto de Código 5.8 *Thread* `SendData`, responsável pelo envio de mensagens entre o `SiRF Interface` e o `IDGPS Server`

```

1  class SendData extends Thread {
2      ...
3      public SendData(connectIDGPS parent) {
4          ...
5      }
6      public void run() {
7          try {
8              sendIdentification();
9              while(!parent.done) {
10                 Thread.sleep(1000);
11                 secCount ++;
12                 if(parent.identified) {
13                     if(parent.iono) {
14                         sendIono();
15                     }
16                     sendCoords();
17                     sendRaw();
18                     sendRTCM();
19                     if(secCount == 300) {
20                         sendEph();
21                         secCount = 0;
22                     }
23                 }
24             }
25         }
26         catch (Exception ex) {
27             ex.printStackTrace();
28         }
29     }
30     ...
31 }

```

A *thread* `ReceiveData`, que escuta a chegada de mensagens provenientes do `IDGPS Server`, encaminha a informação recebida para o método `processMsg()`. Estas mensagens estão definidas no protocolo de comunicação `ProtocolIDGPS`. Este método recebe as respostas aos pedidos de identificação e de fecho de ligação e encaminha-as para os métodos `processMsgIdt()` e `processMsgEnd()`, respectivamente. O extracto de código (5.9) da classe `ReceiveData` apresenta a recepção destas mensagens.

Excerto de Código 5.9 *Thread* `ReceiveData`, responsável pela recepção das respostas do `IDGPS Server`

```

1  class ReceiveData extends Thread {
2      ...
3      public ReceiveData(connectIDGPS parent) {
4          ...
5      }
6      public void run() {
7          try {
8              while (!parent.done) {
9                  processMsg(ProtocolIDGPS.receive(parent.in_data));
10             }
11         } catch (Exception e) {
12             e.printStackTrace();
13         }
14         ...
15     }
16     public void processMsg(ProtocolIDGPS msg) throws Exception {
17         if (msg instanceof MsgIdt) {
18             processMsgIdt((MsgIdt) msg);
19         }
20         else if (msg instanceof MsgEnd) {
21             processMsgEnd((MsgEnd) msg);
22         }
23     }
24     ...
25 }

```

De referir que, no caso desta ligação, não é o servidor quem atribui uma identificação ao cliente, mas é o cliente quem comunica a sua identificação ao servidor. Isto acontece porque a ligação ao `IDGPS Server` só tem sucesso se estiver activa (ou já tiver estado activa) uma ligação ao `EGNOS VRS Server`. Assim, o cliente identifica-se junto do `IDGPS Server` com a mesma identificação que lhe foi atribuída pelo `EGNOS VRS Server`. Exemplificando, se um `SiRF Interface` estiver ligado ao `EGNOS VRS Server` e lhe tenha sido atribuído o número 1 nesta ligação, é-lhe automaticamente atribuída a designação de móvel 101 (ou *rover* 101). Assim, quando este móvel se liga ao `IDGPS Server`, envia no pedido de identificação o conteúdo *rover* 101. Este processo permite identificar os móveis com base nas ligações activas. A Figura (5.9) mostra as mensagens enviadas através da ligação estabelecida entre o `SiRF Interface` e o `IDGPS Server`.

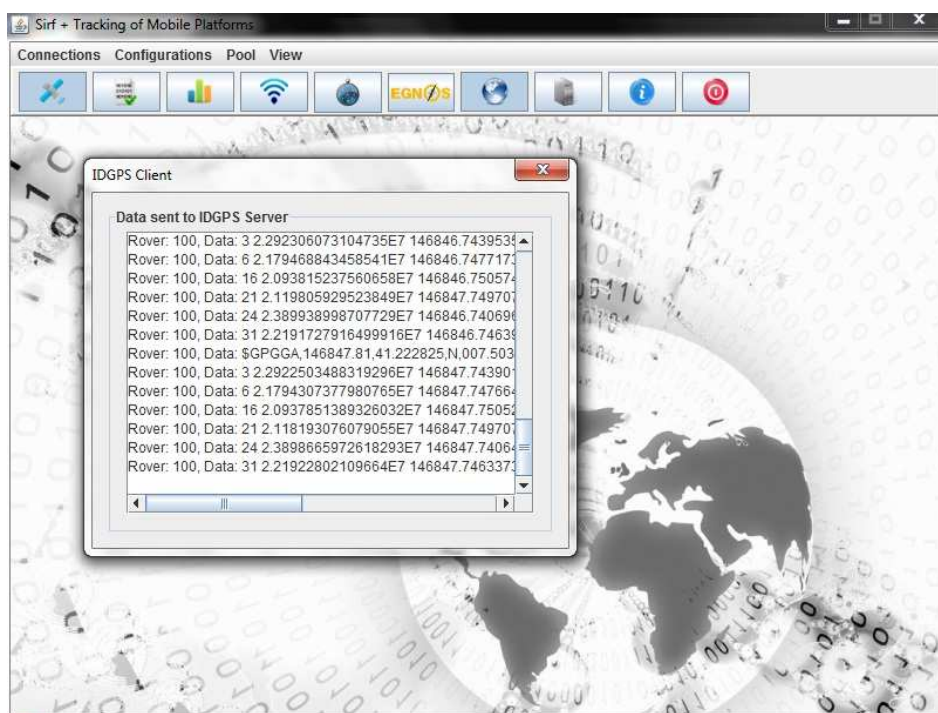


Figura 5.9: SiRF Interface: envio de dados IDGPS Server

5.2.2 Serviço EGNOS VRS

Este módulo é responsável pela aquisição e descodificação do formato RTCA DO-229 das correcções de área alargada do EGNOS e pela implementação para cada cliente do conceito VRS. O conceito VRS implica a recepção da localização aproximada de cada cliente, a geração das correcções específicas que se aplicam a cada local em formato RTCM SC-104 e o seu envio a cada cliente. A sua implementação passa pelos seguintes pontos:

1. Criação de um cliente do serviço SISNeT da ESA que realiza a aquisição, descodificação e tratamento da correcções de área alargada do EGNOS;
2. Determinação das correcções a aplicar às pseudodistâncias dos satélites da constelação com base na informação recebida de cada cliente;
3. Conversão dessa informação para o formato RTCM SC-104;
4. Envio das correcções personalizadas a cada cliente SiRF Interface.

O desenvolvimento deste módulo resulta de uma adaptação da aplicação SISNeTClient/RTCA2RTCM referida no capítulo anterior. Esta aplicação foi reutilizada porque implementa algumas das funcionalidades necessárias, nomeada-

mente a aquisição das correcções do EGNOS e a sua conversão do formato RTCA DO-229 para RTCM SC-104. Apenas de referir que a estrutura gráfica da aplicação `SISNeTClient/RTCA2RTCM` não foi adoptada, tendo sido criada uma nova estrutura capaz de acomodar todas as funcionalidades pretendidas.

5.2.2.1 Aquisição, Descodificação e Tratamento das Correcções de Área Alargada do EGNOS

A execução deste módulo é iniciada pela classe principal `egnosServer`. Esta classe cria a interface gráfica com o utilizador, estabelece a ligação ao SISNeT e prepara o atendimento dos clientes `SiRF Interface`.

Esta aplicação é simultaneamente cliente do serviço SISNeT e servidor dos módulos `SiRF Interface` ligados aos sistema de seguimento de plataformas móveis. O extracto de código (5.10) mostra a criação e inicialização das duas classes que suportam as funcionalidades descritas: a `SISNeTClient` e a `VRSServer`.

Excerto de Código 5.10 Método construtor da classe principal `egnosServer`

```

1  public class egnosServer extends javax.swing.JFrame {
2      public SISNeTClient client = null;
3      public VRSServer server = null;
4      ...
5      public egnosServer() {
6          initComponents();
7          ...
8          this.client = new SISNeTClient(this);
9          this.client.rov.setCoords(new EcefRecord(4753502.000, -719543.000, 4177535.000));
10         this.client.fullCor_enabled = true;
11         this.server = new VRSServer(this);
12         this.server.openLog();
13         this.server.start();
14         ...
15     }
16     ...
17 }

```

O cliente SISNeT é iniciado pela classe `SISNeTClient` que, quando executada, cria um objecto `GPSConstellation` com a capacidade para comportar informação relativa a 37 satélites. Uma vez estabelecida a ligação, procede-se à autenticação. Caso seja sucedida, dá-se início ao pedido de mensagens SISNeT. O pedido de mensagens é feito através da classe `Requester`, que após a autenticação junto do SISNeT, inicia uma *thread* que requisita as mensagens periodicamente, segundo a seguinte ordem:

- As correcções ionosféricas, nomeadamente, os parâmetros do modelo ionosférico;
- As efemérides de cada satélite da constelação NAVSTAR-GPS;

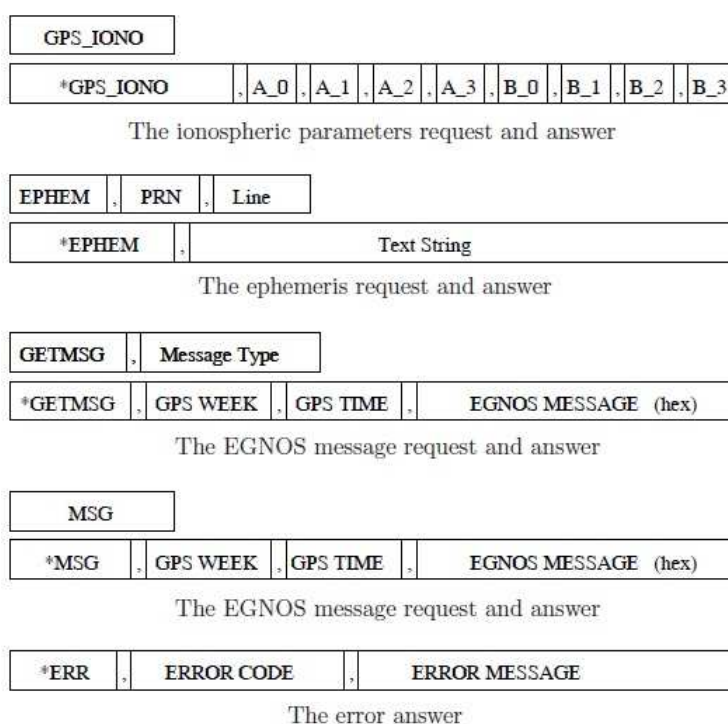


Figura 5.10: SISNeT: mensagens utilizadas

- As mais recentes mensagens RAÇA DO-229 MT1, MT18, MT2, MT3, MT4 e MT5;
- A última mensagem difundida pelo EGNOS.

A Figura (5.10) mostra as mensagens que são requisitadas ao SISNeT e as respectivas respostas.

O protocolo de comunicação definido entre o EGNOS VRS Server e o serviço SISNeT está implementado através da classe `Protocol`. A escuta de respostas é processada pelo método `processAnswer()` da classe `SISNeTClient`. O extracto de código (5.11) apresenta este método, nomeadamente as acções que são desencadeadas aquando da chegada dos diferentes tipos de mensagens.

Após a descodificação das mensagens RTCA DO-229, a informação extraída é armazenada numa instância da classe `GPSConstellation`. A `GPSConstellation` define uma estrutura de dados para comportar a informação relativa a cada um dos satélites da constelação NAVSTAR-GPS, sendo cada satélite representado por um objecto `GPSSatellite`. Cada objecto `GPSSatellite` armazena a informação relativa às efemérides, correcções rápidas e de longo prazo, correcções ionosféricas e troposféricas, *etc.* de um satélite. A descodificação das mensagens RTCA DO-229 é realizada pela classe `DataBlock` que armazena cada mensagem num objecto

Excerto de Código 5.11 Método `processAnswer()` da classe `SISNetClient` que escuta a chegada de mensagens provenientes do SISNeT

```

1  public class SISNetClient implements Runnable {
2      ...
3      public void run() {
4          int input_byte = 0;
5          try {
6              ...
7              while (work) {
8                  input_byte = sisnet_in.read();
9                  switch (input_byte) {
10                     case -1:
11                         break;
12                     case 10:
13                         answer.append((char) input_byte);
14                         processAnswer(answer.toString());
15                         answer = new StringBuffer();
16                         break;
17                     case 13:
18                         break;
19                     default:
20                         answer.append((char) input_byte);
21                 }
22             }
23         } catch (Exception e) {
24             e.printStackTrace();
25             setWork(false);
26         }
27     }
28     ...
29     public void processAnswer(String msg) {
30         if (msg.startsWith("*ERR")) {
31             this.egnos.displayDump(" + new ErrAns(msg) + ");
32             return;
33         }
34         if (msg.startsWith("*TXT")) {
35             this.egnos.displayDump(" + new TxtAns(msg) + ");
36             return;
37         }
38         if (msg.startsWith("*GETMSG")) {
39             ans = new MsgAns(msg);
40             data = new DataBlock(ans.gps_time, ans.egnos, this);
41             this.egnos.displayDump(" + ans + ");
42             return;
43         }
44         ...
45     }
46 }

```

do tipo `BitSet` de 250 b de comprimento. Posteriormente, é invocado o método `process()`, que extrai os primeiros 8 b do `BitSet` (ou seja, o preâmbulo da mensagem) e descodifica a restante informação mediante o tipo de mensagem (o que é feito através das classes `getPreamble()`, `getMT()` e `getAppMsg()`). Este processo encontra-se ilustrado no extracto de código (5.12). As mensagens a descodificar pela aplicação são [23]:

Mensagem Tipo 1 (MT1) Deve ser descodificada antes de qualquer outra mensagem EGNOS porque as mensagens 2-5, 24, 25, 27 e 28 dependem do conteúdo desta mensagem para serem devidamente descodificadas. Esta mensagem contém a definição das máscaras PRN dos satélites seguidos, *i.e.*, indica quais os satélites das constelações GPS, GLONASS e Galileo es-

Excerto de Código 5.12 Classe `DataBlock`, que descodifica a informação recebida do serviço SISNeT no formato RTCA DO-229

```

1  public class DataBlock {
2      BitSet block = null;
3      SISNeTClient parent = null;
4      public double messageTime;
5      public int preamble;
6      public int message_type;
7      ...
8      public DataBlock(double gps_time, String egnos, SISNeTClient p) {
9          parent = p;
10         messageTime = gps_time;
11         block = new BitSet(252);
12         ...
13         process();
14     }
15     public void process() {
16         preamble = getPreamble();
17         message_type = getMT();
18         getAppMsg(message_type);
19     }
20     public int getPreamble() {
21         ...
22     }
23     public int getMT() {
24         ...
25     }
26     public void getAppMsg(int mt) {
27         switch (mt) {
28             case 0:
29                 decodeMT_00();
30                 decodeMT_02Till105(2);
31                 break;
32             case 1:
33                 decodeMT_01();
34                 break;
35             ...
36         }
37     }
38     public void decodeMT_02Till105(int mt) {
39         ...
40     }
41     public void decodeMT_00() {
42         ...
43     }
44     public void decodeMT_01() {
45         ...
46     }
47     ...
48 }

```

tão a ser monitorados pelo sistema. Cada satélite tem a sua posição nesta máscara e quando o sistema coloca o bit correspondente a **true** ou **false**, indica que está ou não a seguir esse satélite. A `DataBlock` lê esta informação e, para cada objecto `GPSSatellite`, inicia o campo que indica se existe informação acerca do satélite de acordo com o valor correspondente da máscara PRN. Esta mensagem pode conter a informação das máscaras PRN de 210 satélites.

Mensagens Tipo 2 a 5 (MT2-5) As mensagens 2 a 5 são muito semelhantes e contêm as correcções rápidas e o URDE de 13 satélites. A mensagem 2 contém a informação dos primeiros 13 satélites da máscara PRN, a mensagem 3 contém a informação dos satélites 14-26 e assim sucessivamente. A

mensagem de correções rápidas apenas é difundida se o número de satélites descritos na máscara PRN o justificar, isto é, a mensagem 5 só será difundida se 40 ou mais satélites estiverem descritos na MT1. Segundo os criadores da *SISNetClient/RTCA2RTCM*, isto nunca aconteceu durante a fase de testes, logo a MT5 não foi implementada [23].

Mensagem Tipo 6 (MT6) Contém a informação de integridade do EGNOS, sob a forma de 51 valores DUREI que são usados para determinar a variação do URDE, indicando a exactidão do valor combinado das correções rápidas e das as correções de longo prazo (não incluindo a exactidão das correções ionosféricas dada pela MT26).

Mensagem Tipo 7 (MT7) Contém o factor de degradação das correções rápidas para todos os satélites designados na máscara PRN. Estes factores de degradação permitem definir durante quanto tempo as correções rápidas de um satélite são válidas. Por exemplo, se o factor de degradação de uma correção rápida for 18 e a correção rápida de um determinado satélite não for actualizado durante 18 s, estes dados não devem ser utilizados.

Mensagem Tipo 9 (MT9) Contém a posição, velocidade, aceleração, desvio de relógio e desvios de frequência do satélite geostacionário que está a difundir a informação do EGNOS, bem como o tempo de aplicabilidade t_0 e o expoente da exactidão (URA). Estes valores são descodificados pela aplicação, mas não são utilizados em qualquer cálculo.

Mensagem Tipo 10 (MT10) Os factores de degradação são descodificados pela aplicação, mas não são utilizados em qualquer cálculo.

Mensagem Tipo 12 (MT12) Os parâmetros temporais relativos ao WAAS *Network Time/UTC/GLONASS Time Offset* são descodificados pela aplicação, mas não são utilizados em qualquer cálculo.

Mensagem Tipo 17 (MT17) Os almanaques dos satélites geostacionários usados pelo EGNOS são difundidos periodicamente para alertar acerca da existência, localização, estado de saúde e estado geral dos três satélites geostacionários utilizados. Os almanaques inutilizados têm um PRN igual a 0 e devem ser ignorados. Esta informação é descodificada, mas não é utilizada em qualquer cálculo.

Mensagem Tipo 18 (MT18) Contém informação acerca dos 201 pontos da grelha ionosfera - *Ionospheric Grid Points (IGP)* - de uma única faixa da grelha ionosfera definida pelo EGNOS. Consiste numa máscara de pontos denominada *ionospheric grid mask* ou máscara da grelha ionosfera que indica quais os IGP com correção ionosfera disponível. Adicionalmente,

também contém um indicador que indica quantas faixas estão a ser difundidas pelo serviço. A máscara da grelha ionosfera é representada pela classe `IonosphericGridWorld` e cada faixa corresponde a um objecto `IonosphericGridBand`.

Mensagem Tipo 24 (MT24) Esta mensagem contém correcções rápidas e correcções de longo prazo, sendo uma combinação das mensagens 2-5 e da mensagem 25. A primeira metade desta mensagem contém as correcções rápidas de 6 satélites e a segunda metade contém metade de uma mensagem 25.

Mensagem Tipo 25 (MT25) Contém as correcções de longo prazo e é difundida para fornecer as estimativas de erro para as efemérides e desvios de relógio relativamente às coordenadas ECEF do WGS 84. Estas correcções são estimadas relativamente às efemérides e desvios de relógio difundidas pelo GNSS. A descodificação desta mensagem depende de um valor booleano, também contido nesta mensagem, que indica se é possível descodificar correcções de longo prazo de 4 satélites (se for 0) ou de 2 satélites (se for 1). O satélite ao qual corresponde a informação descodificada é identificado pelo número de acordo com a sua posição na máscara PRN. As correcções de longo prazo de um satélite são armazenadas num objecto da classe `LongTermCorrection`.

Mensagem Tipo 26 (MT26) Fornece os atrasos verticais e a exactidão (σ_{GIVE}^2) dos IGP. A informação é agrupada em blocos de 15 pontos, sendo a localização de cada bloco na faixa respectiva identificado por um *block ID*. O bloco 0 contém as correcções para os primeiros 15 IGP da máscara da grelha ionosfera, o bloco 1 contém as correcções dos IGP 16 a 30 e assim sucessivamente. A informação de cada IGP é armazenada num objecto `IonosphericGridPoint`.

O conteúdo descodificado das mensagens RTCA DO-229 é disponibilizado na interface gráfica através de diversas tabelas como se pode ver na Figura (5.11). As mensagens do SISNeT (pedidos e respostas) são também apresentadas noutras duas áreas como se apresenta na Figura (5.12).

5.2.2.2 Correcções das Pseudodistâncias

Uma vez de posse de todas as mensagens enviadas pelo SISNeT, é possível proceder ao cálculo das correcções a aplicar às pseudodistâncias brutas medidas por cada receptor. Essa informação é essencial para a conversão do formato RTCA DO-229 para RTCM SC-104, *i.e.*, a transformação das correcções de área alargada em correcções diferenciais específicas de cada receptor GNSS.

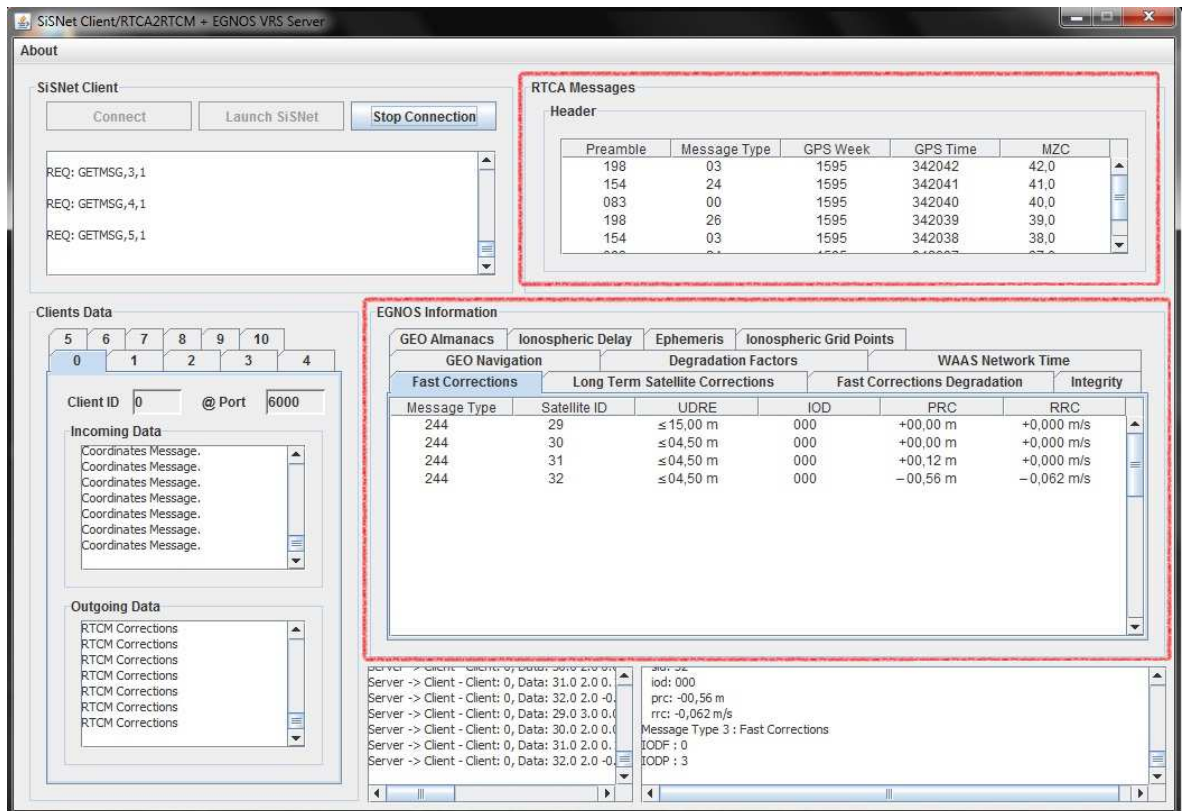


Figura 5.11: EGNOS VRS Server: decodificação das mensagens RTCA DO-229

A determinação da correcção total a aplicar à pseudodistância de um satélite utiliza os seguintes parâmetros:

- Correcções rápidas contidas nas MT2-5 e MT24;
- Informação de integridade e factores de degradação das correcções rápidas contidas nas MT6 e MT7;
- Correcções de longo prazo contidas na MT25;
- Correcções troposféricas que não são disponibilizadas pelo SISNeT e que têm de ser determinadas analiticamente;
- Correcções ionosféricas contidas nas MT18 e MT26 e que devem ser aplicadas à localização geográfica do receptor.

Correcções Rápidas - As correcções rápidas estão contidas nas MT2-5 e MT24 e são complementadas pelas MT6 e MT7, que possuem informação de integridade e factores de degradação das correcções rápidas. A MT6 contém

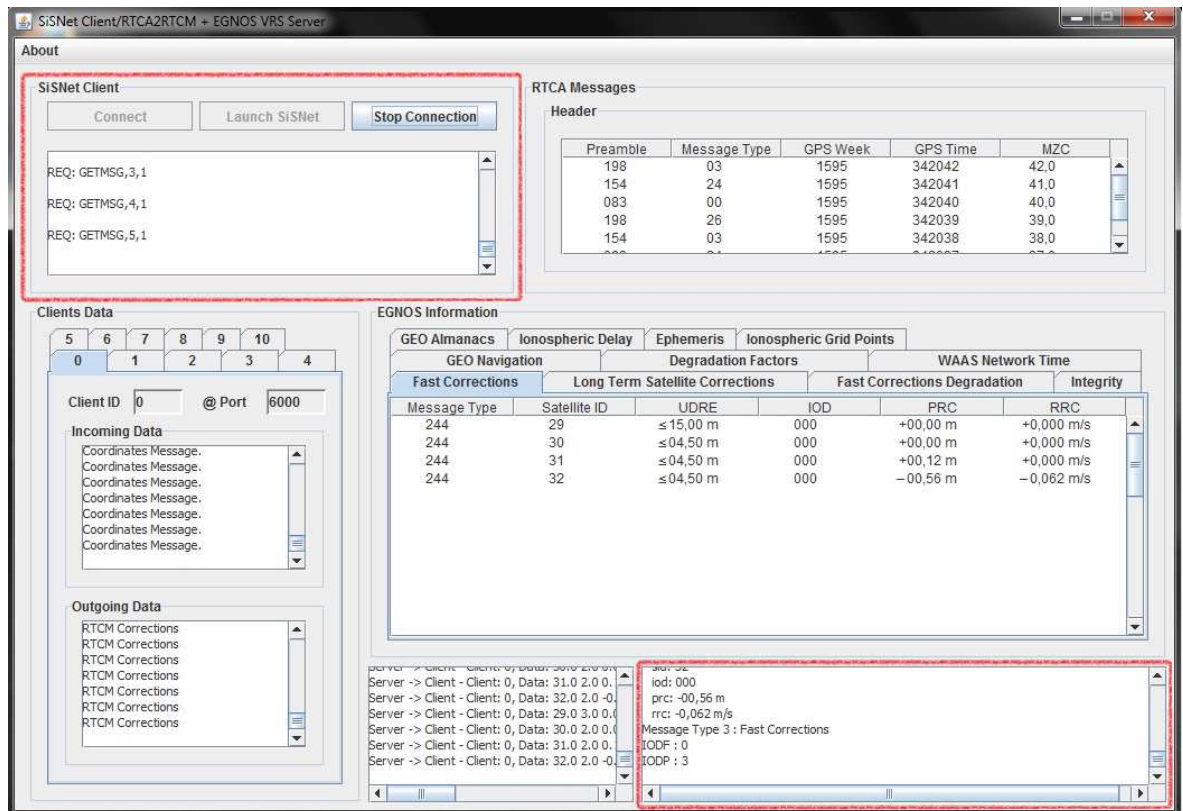


Figura 5.12: EGNOS VRS Server: interação com o SISNet

os valores DUREI e quatro IODA relativos a cada mensagem 2 a 5. A MT7 contém os factores de degradação que devem ser aplicados aos valores DUREI da mensagem 6.

As correcções rápidas de um satélite devem ser processadas analiticamente antes de serem aplicadas. Os valores contidos nas MT2-5 correspondem à correcção da pseudodistância (PRC) e ao URDE do satélite em questão. Assim e em primeiro lugar, tem de se calcular a taxa de variação da PRC que é dada pela Equação (5.42).

Equação 5.42: Correcção rápida da *range-rate*

$$RRC(t_{of}) = \frac{PRC_{actual} - PRC_{anterior}}{t_{of,actual} - t_{of,anterior}} \quad (m/s) \quad (5.42)$$

onde $RRC(t_{of})$ é a correcção *range-rate*, PRC_{actual} é a correcção da pseudodistância actual, $PRC_{anterior}$ é o valor da correcção da pseudodistância imediatamente anterior, $t_{of,actual}$ é o tempo de aplicabilidade da PRC actual

e $t_{of,anterior}$ é o tempo de aplicabilidade da PRC imediatamente anterior. Foram ainda levadas em conta algumas considerações adicionais relativamente à RRC que estão detalhadas em [13].

Deste modo, é possível calcular a correcção da distância com base nas correcções rápidas através da Equação (5.43).

Equação 5.43: Correcção rápida da distância total

$$RC_{fast}(t) = PRC(t_{of}) + RRC(t_{of})(t - t_{of}) \quad (m) \quad (5.43)$$

em que $RC_{fast}(t)$ é a correcção da distância total, t_{of} é o tempo de aplicabilidade das correcções rápidas actuais e t é tempo actual.

Correcções de Longo Prazo - As correcções de longo prazo contidas na MT25 incluem as correcções relativas à variação lenta das efemérides (correcções à posição ECEF e respectiva taxa de variação) e dos desvios de relógio dos satélites (desvio e taxa de variação). A MT25 também contém um IDO associado que, em conjunto com o IODE das efemérides, informa acerca de quando e como devem ser aplicadas as correcções de longo prazo. Podem ocorrer três situações:

1. Se o IDO e o IODE são iguais, a correcção é aplicada utilizando as efemérides actuais;
2. Se o IDO é diferente do IODE mas é igual ao IODE das efemérides anteriores, a correcção é aplicada usando as efemérides anteriores (o que não acontece visto não existir um *buffer* criado para manter estas efemérides);
3. Se o IDO for diferente de qualquer IODE recebido, não podem ser aplicadas quaisquer correcções (rápidas ou de longo prazo).

Adicionalmente, pode ocorrer um *timeout*, *i.e.*, se a última MT25 tiver sido recebida há mais de 240 s, os dados estão desactualizados e não podem ser aplicados.

As correcções de longo prazo da posição de um satélite são dadas pela Equação (5.44).

Equação 5.44: Correcção de longo prazo da posição do satélite

$$\begin{aligned} DX(t) &= DX + DX_{ROC}(t - t_0) \quad (m) \\ DY(t) &= DY + DY_{ROC}(t - t_0) \quad (m) \\ DZ(t) &= DZ + DZ_{ROC}(t - t_0) \quad (m) \end{aligned} \quad (5.44)$$

onde $DX(t)$, $DY(t)$ e $DZ(t)$ são as correcções à posição actual do satélite, DX , DY e DZ são as correcções à posição do satélite transmitidas, DX_{ROC} , DY_{ROC} e DZ_{ROC} são as correcções à velocidade do satélite transmitidas, t é o tempo actual e t_0 é o tempo de aplicabilidade das correcções à velocidade. A correcção de relógio de um satélite é dada pela Equação (5.45).

Equação 5.45: Correcção de longo prazo do relógio do satélite

$$\delta\Delta_{t_{SV}}(t) = \delta_{a_{f0}} + \delta_{a_{f1}}(t - t_0) \quad (s) \quad (5.45)$$

onde $\delta\Delta_{t_{SV}}(t)$ é a correcção de relógio resultante, $\delta_{a_{f0}}$ e $\delta_{a_{f1}}$ são os parâmetros de correcção de relógio contidos na MT25, t_0 é o tempo de aplicabilidade dos parâmetros de correcção de relógio e t é o tempo actual. Assim, quando a correcção de relógio $\delta\Delta_{t_{SV}}(t)$ for conhecida, é possível calcular a correspondente correcção à distância através da Equação (5.46).

Equação 5.46: Componente do relógio do satélite de correcção da distância

$$RC_{clock} = \delta\Delta_{t_{SV}}(t) * c \quad (m) \quad (5.46)$$

em que RC_{clock} é a correcção resultante, $\delta\Delta_{t_{SV}}(t)$ é a correcção de relógio e c é a velocidade da luz no vazio ($2,99792458 \times 10^8$ m/s).

Correcções Troposféricas - Apesar do EGNOS não providenciar correcções troposféricas, a especificação RAÇA DO-229 recomenda o uso de um algoritmo para estimar o atraso troposférico baseado na latitude e longitude do receptor e na pressão, temperatura e vapor de água [11]. O algoritmo utilizado é descrito em [13] e resulta na correcção troposférica dada pela Equação (5.47).

Equação 5.47: Componente troposférica de correcção da distância

$$RC_{tropo} = -(d_{hyd} + d_{wet}) * \frac{1,001}{\sqrt{0,002001 + \sin E_i^2}} \quad (m) \quad (5.47)$$

em que RC_{tropo} é a correcção troposférica, d_{hyd} e d_{wet} são as componentes húmida e seca descritas em [13] e E_i é a elevação do satélite i em questão.

Correcções Ionosféricas - Segundo a especificação RAÇA DO-229, é possível determinar o atraso ionosférico que afecta o sinal de um satélite conhecendo o seu ponto de perfuração da ionosfera ou *Ionospheric Pierce Point* (IPP). Traçando uma recta imaginária entre a posição do receptor e a posição

do satélite em questão, o IPP é o ponto dessa trajectória que intersecta a camada ionosfera como se ilustra na Figura (5.13). As coordenadas deste ponto são determinadas analiticamente com base num conjunto de equações descritas em [13].

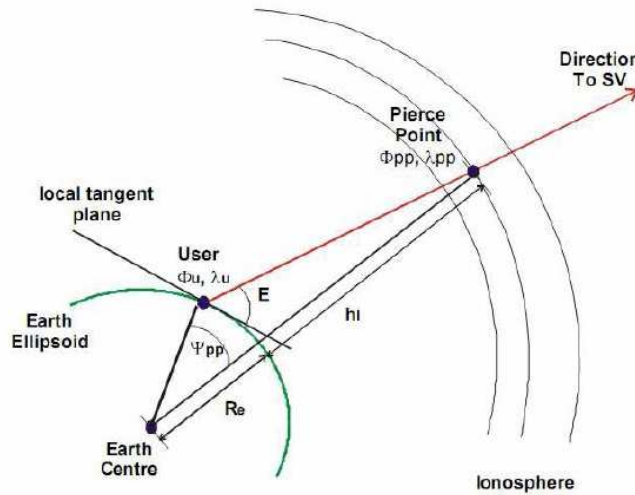


Figura 5.13: IPP em relação a um receptor e a um satélite [13]

As MT18 e MT25 contêm informação acerca do atraso ionosférico de pontos geodésicos específicos: os IGP. Estes pontos cobrem a superfície terrestre ao longo de 11 faixas, sendo cada faixa constituída por 201 pontos como ilustra a Figura (5.14). Deste modo, os IGP criam uma grelha de pontos com posições exactas e cujo atraso ionosférico é conhecido. Conhecendo esta grelha e as coordenadas do IPP do satélite em questão, é possível interpolar o atraso ionosférico nesse IPP.

Dependendo da latitude do IPP, a especificação RTCA DO-229 sugere diversas abordagens relativamente à forma de realizar esta interpolação. Os diferentes tipos de interpolação dependem do número de IGP com atraso ionosférico disponível (só os IGP monitorados possuem valor para o atraso ionosférico). Existem diferentes abordagens de cálculo do atraso de um IPP. Estas abordagens, que são baseadas em formas celulares descritas em [13], implementam uma interpolação baseada na distância entre o IPP e os IGP circundantes. Por fim, o atraso ionosférico no IPP é ajustado através de um factor de obliquidade, obtendo-se a correcção ionosfera para a pseudodistância do satélite em questão através da Equação (5.48).

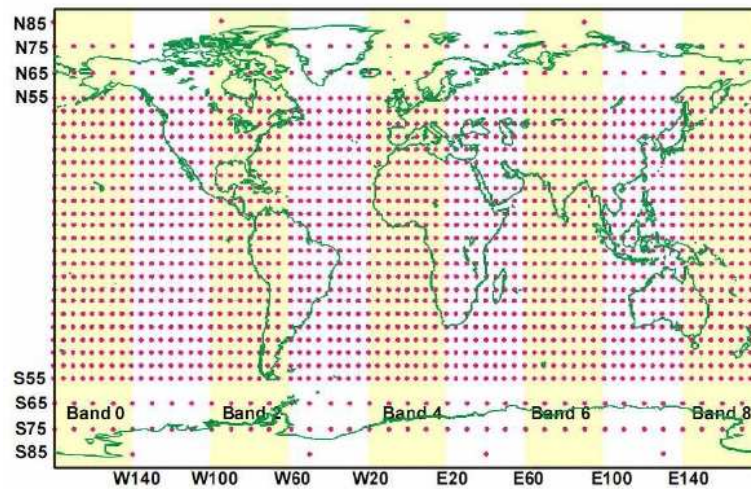


Figura 5.14: Pontos da grade ionosférica e respectivas faixas [13]

Equação 5.48: Componente ionosférica de correção da distância

$$RC_{iono} = -F_{pp} * Ionospheric\ Delay \quad (m) \quad (5.48)$$

onde RC_{iono} é a correção ionosférica, F_{pp} é o factor de obliquidade e $Ionospheric\ Delay$ é o atraso ionosférico calculado através da interpolação.

Finalmente, é possível calcular a correção final a aplicar à pseudodistância de cada satélite recorrendo à Equação (5.49).

Equação 5.49: VRS: Correção da pseudodistância

$$PRC_{vrs} = RC_{fast} - RC_{iono} + RC_{tropo} + RC_{clock} \quad (m) \quad (5.49)$$

em que PRC_{vrs} é a correção da pseudodistância segundo o conceito VRS, RC_{fast} é a correção com base nas correções rápidas, RC_{iono} é a correção ionosférica, RC_{tropo} é a correção troposférica e RC_{clock} é correção de relógio do satélite. Este cálculo é efectuado pelo método `calculatePRC()` da classe `GPSSatellite`, uma vez determinadas todas as correções aplicáveis.

5.2.2.3 Conversão de RTCA DO-229 para RTCM SC-104

Uma das funcionalidades do módulo `EGNOS VRS Server` é a conversão das mensagens do formato RTCA DO-229 para o formato RTCM SC-104. Como se trata

de uma conversão em tempo útil, pressupõe-se que deve ser feita sempre que é recebida do SISNeT uma mensagem RTCA DO-229.

Para o efeito, é utilizado o método `generateRTCM()`, invocado pela classe `SISNeTClient` no método `processAnswer()`, que é o método que escuta a chegada de mensagens provenientes do SISNeT. O extracto de código mostra a invocação do método `generateRTCM()` que é executado sempre que se assinala a chegada de uma mensagem EGNOS proveniente do SISNeT.

Excerto de Código 5.13 Chegada de uma mensagem EGNOS e invocação do método `generateRTCM()`

```

1  public void processAnswer(String msg) {
2      ...
3      if (msg.startsWith("MSG")) {
4          ans = new MsgAns(msg);
5          this.egnos.displayDump(" " + ans + " ");
6          generateRTCM();
7          return;
8      }
9      ...
10 }

```

Este método é responsável pela extracção e armazenamento do conteúdo das mensagens RTCA DO-229 nos objectos da classe `GPSSatellite` correspondentes aos satélites monitorados pelo EGNOS. Com base nesta informação e na localização fornecida¹, determinam-se as correcções a aplicar às pseudodistâncias observadas. Desta forma, transformam-se as correcções de área alargada, dispersas por várias mensagens RTCA DO-229, em correcções diferenciais específicas de um dado local. Por fim, é invocado o método `makeRtcmMessage()` que cria o bloco de informação em formato RTCM SC-104 com os dados diferenciais calculados e recebe como argumento o tipo de mensagem RTCM SC-104 a gerar. Todo este processo está descrito no extracto de código (5.14).

O método `makeRtcmMessage()` gera dois tipos de mensagens RTCM SC-104:

- MT1 *Differential GPS Corrections* - contém as correcções diferenciais de todos os satélites visíveis;
- MT3 *GPS Reference Station Parameters* - contém as coordenadas da estação de referência virtual criada.

Quando invocado, este método cria o cabeçalho da mensagem RTCM SC-104, que é comum a qualquer tipo de mensagem, através da criação de um objecto `RtcmHeader`. O cabeçalho é então preenchido com o tipo de mensagem, o tamanho da mensagem, o identificador da estação que está a gerar as correcções

¹Por omissão, *i.e.*, quando não tem clientes ligados, o `VRS Server` gera correcções diferenciais para um ponto localizado no terraço do edifício F do campus do ISEP.

Excerto de Código 5.14 Método `generateRTCM()` que cria as mensagens RTCM SC-104 com as correcções diferenciais

```

1  public void generateRTCM (){
2      for (int i = 0; i < gps_const.dimSats; i++) {
3          if (gps_const.satellite[i].ephemeris.svid != 0
4              && gps_const.satellite[i].ephemeris.toc_str != null) {
5              if (gps_const.satellite[i].ephemeris.iode ==
6                  gps_const.satellite[i].longtermcorrection.IOPD
7                  && gps_const.satellite[i].ephemeris.iodc ==
8                  gps_const.satellite[i].longtermcorrection.IOPD) {
9                  gps_const.satellite[i].longtermcorrection.correct(
10                     ans.gps_time - (ans.gps_day * 60 * 60 * 24));
11              }
12              else {
13                  gps_const.satellite[i].longtermcorrection.corrAvailable = false;
14              }
15              gps_const.satellite[i].Ipp.setCords(gps_const.satellite[i],
16                  this.rov, ans.gps_time, (ans.gps_day * 60 * 60 * 24));
17              gps_const.satellite[i].Ipp.fillGrid();
18              gps_const.satellite[i].Ipp.resolveShape(
19                  gps_const.satellite[i].Ipp.Grid);
20              gps_const.satellite[i].Ipp.interpolate(ans.gps_time);
21              gps_const.satellite[i].Tropo.Calculate(gps_const.satellite[i], this.rov);
22              if (gps_const.satellite[i].ephemeris.iode ==
23                  gps_const.satellite[i].longtermcorrection.IOPD
24                  && gps_const.satellite[i].ephemeris.iodc ==
25                  gps_const.satellite[i].longtermcorrection.IOPD) {
26                  gps_const.satellite[i].longtermcorrection.correct(
27                     ans.gps_time - (ans.gps_day * 60 * 60 * 24));
28              }
29              else {
30                  gps_const.satellite[i].longtermcorrection.corrAvailable = false;
31              }
32          }
33      }
34      ...
35      counter++;
36      makeRtcmMessage(1);
37      if (counter == 30) {
38          makeRtcmMessage(3);
39          counter %= 30;
40      }
41  }

```

diferenciais, o estado de saúde (calculado através do URDE das correcções rápidas RTCA DO-229 e segundo uma tabela definida na especificação RTCM SC-104 [12]) e o parâmetro *modified z-count*.

Posteriormente, é criado um novo objecto para comportar o corpo da mensagem através da classe `RtcmMsg`. Dependendo do tipo de mensagem a gerar, são invocados métodos distintos. No caso de se pretender gerar uma MT1, é invocado o método `create_01()`; no caso de se desejar criar uma MT3, é invocado o método `combine_03()`. Estes métodos realizam as seguintes funções:

- O método `create_01()` cria uma instância da classe `RtcmType1` para cada satélite da constelação com o factor de escala, o intervalo entre os URDE, o identificador do satélite, a correcção da pseudodistância, a correcção *range-rate* e o IDO da mensagem. Estes parâmetros são os campos de uma MT1 RTCM SC-104[12].

- O método `combine_03()` cria uma instância da classe `RtcmType3` contendo as coordenadas da estação virtual que gerou as correcções. Esta mensagem é criada obedecendo à especificação das MT3 RTCM SC-104 [12].

Por fim, o corpo e cabeçalho da mensagem gerada são codificados através dos métodos `encodeHeader()` e `encodeBody()`.

O método `makeRtcmMessage()` encontra-se descrito no extracto de código (5.15), onde são visíveis os passos necessários à construção da mensagem.

Excerto de Código 5.15 Método `makeRtcmMsg()` que cria as mensagens RTCM SC-104 MT1 e MT3

```

1  public void makeRtcmMessage(int MT) {
2      boolean D29 = false;
3      boolean D30 = false;
4      int id = 0;
5      int sequence = 0;
6      ...
7      RtcmHeader header = new RtcmHeader();
8      switch (MT){
9          case 1:
10             header.setType(1);
11             header.setId(REF_STATION_ID);
12             header.setMzc(ans.gps_time / 3600);
13             header.setSequence(sequence);
14             howManySats = 0;
15             for (int i = 1; i <= 36; i++) {
16                 if (gps_const.satellite[i].prnFlag
17                     && !gps_const.satellite[i].fastcorrection.udre.equals("Not Monitored")
18                     && !gps_const.satellite[i].fastcorrection.udre.equals("Do Not Use")
19                     && !gps_const.satellite[i].fastcorrection.udre.equals("ERROR")
20                     && Double.parseDouble(gps_const.satellite[i].fastcorrection.udre) < 150) {
21                     howManySats++;
22                 }
23             }
24             header.setLength((int) Math.ceil(((double) 40 / 24) * (double) howManySats));
25             this.egnos.displayDump("How many satellites : " + howManySats);
26             break;
27          case 3:
28             header.setType(3);
29             header.setId(REF_STATION_ID);
30             header.setMzc(ans.gps_time / 3600);
31             header.setSequence(sequence);
32             header.setLength(4);
33             break;
34         }
35         int health = determineUdreRtcm();
36         header.setHealth(health);
37         new_message = new RtcmMsg();
38         new_message.setHeader(header);
39         new_message.setId(id);
40         new_message.setD29(D29);
41         new_message.setD30(D30);
42         if (MT == 1) {
43             new_message.create_01(gps_const, ans, health, fullCor_enabled);
44         }
45         if (MT == 3) {
46             new_message.combine_03(rov);
47         }
48         new_message.encodeHeader();
49         new_message.encodeBody();
50         ...
51     }

```

5.2.2.4 Estação de Referência Virtual

Tal como foi referido anteriormente, o módulo **EGNOS VRS Server**, além de ser cliente do serviço **SISNeT**, é um servidor de correcções diferenciais geradas de acordo com o conceito **VRS**. Nesta qualidade, o **EGNOS VRS Server** presta os seguintes serviços aos seus clientes **SIRF Interface**:

- Recepção da posição aproximada dos clientes;
- Envio periódico das efemérides da constelação NAVSTAR-GPS recebidas do **SISNeT**;
- Envio periódico das correcções diferenciais em formato RTCM SC-104 geradas segundo o conceito **VRS**.

O servidor, que é lançado pela classe **VRSServer** no porto 5060, tem a capacidade de atender em simultâneo até 10 clientes². A chegada de um cliente desencadeia a criação de uma *thread* de atendimento do tipo **HandleClient** que é responsável pela troca de mensagens com o cliente em questão. A criação do *socket* de escuta e o lançamento desta *thread* são exemplificados no extracto de código (5.16).

A *thread* **HandleClient** processa a recepção e envio das mensagens da seguinte forma:

- Quando recebe o pedido inicial de identificação do cliente, responde com uma mensagem contendo um identificador único;
- Sempre que recebe uma mensagem com as coordenadas do cliente, gera e envia uma mensagem com as correcções diferenciais específicas no formato RTCM SC-104;
- Enquanto o cliente estiver ligado, envia a cada 5 min uma mensagem contendo as efemérides mais recentes recebidas do **SISNeT**;
- Quando recebe o pedido de fecho de ligação do cliente, responde com uma mensagem de anuência.

O método **processMsg()** é responsável pelo processamento das mensagens dos clientes definidas na classe **ProtocolEGNOS**:

- Os pedidos de identificação são encaminhados para o método **processMsgIdt()**;

²Este número de clientes permitiu realizar a prova de conceito, não constituindo uma limitação do sistema.

Excerto de Código 5.16 Método construtor da classe `VRSServer` e criação das *threads* `HandleClient` de atendimento dos clientes `SiRF Interface`

```

1  public class VRSServer extends Thread {
2      private final int dim_clients = 10;
3      private ServerSocket socket_server = null;
4      private Vector clients = null;
5      public int port = 5060;
6      ...
7      public VRSServer(egnosServer parent) {
8          ...
9          clients = new Vector();
10         try {
11             socket_server = new ServerSocket(port, dim_clients);
12             this.parent.displayClients("VRS Server initiated!");
13             this.parent.displayClients("Listening to clients on port "+port+".");
14         }
15         catch (IOException e) {
16             e.printStackTrace();
17             this.parent.displayClients("Failed to initiate VRS Server!");
18         }
19     }
20     public void run() {
21         try {
22             while(true) {
23                 if(clients.size() < dim_clients) {
24                     clients.addElement(new HandleClient(socket_server.accept(), this, clients.size()));
25                     ((HandleClient)clients.lastElement()).start();
26                     this.parent.displayClients("Client "+(clients.size()-1)+" connected!");
27                 }
28                 else if(clients.size() == dim_clients) {
29                     this.parent.displayClients("Maximum number of clients exceeded!");
30                     Thread.sleep(1000);
31                     this.parent.displayClients("Checking if any client has disconnected itself...");
32                 }
33             }
34         }
35         catch (Exception e) {
36             e.printStackTrace();
37             System.exit(1);
38         }
39     }
40     ...
41 }

```

- As mensagens com as coordenadas em formato NMEA são encaminhadas para o método `processMsgPos()`;
- Os pedidos de fecho da ligação estabelecida são encaminhados para o método `processMsgEnd()`.

A escuta e processamento das mensagens do cliente encontra-se descrita no extracto de código (5.17).

Um dos processos mais importantes desta parte do módulo `EGNOS VRS Server` é, precisamente, a determinação da correcções a aplicar às pseudodistâncias dos satélites da constelação em função da posição de cada cliente `SiRF Interface` ligado ao servidor. Tal como foi referido, isto é feito através da extracção da posição do cliente em questão, que é enviada de segundo a segundo em formato NMEA, seguida da chamada ao método `generateClientRTCM()` da classe `SISNeTClient`. Este método, descrito no extracto de código (5.18), é muito semelhante ao método `generateRTCM()`: extrai as mais recentes correcções de área

Excerto de Código 5.17 Escuta e processamento das mensagens na classe `HandleClient` do `EGNOS VRS Server`

```

1  public void run() {
2      ...
3      try {
4          while(!done) {
5              processMsg(ProtocolEGNOS.receive(in_data));
6              Thread.sleep(1000);
7              this.secCount ++;
8              if(secCount == 300) {
9                  sendEph();
10                 secCount = 0;
11             }
12         }
13         Thread.sleep(1500);
14         conn.close();
15         this.server.parent.displayClients("Socket closed!");
16     }
17     catch(Exception e) {
18         e.printStackTrace();
19     }
20 }
21 public void processMsg(ProtocolEGNOS msg) throws Exception {
22     if(msg instanceof MsgIdt)
23         processMsgIdt((MsgIdt)msg);
24     else if(msg instanceof MsgPos)
25         processMsgPos((MsgPos)msg);
26     else if(msg instanceof MsgEnd)
27         processMsgEnd((MsgEnd)msg);
28 }

```

alargada das mensagens RTCA DO-229 e efectua os cálculos necessários à determinação da correcção final à pseudodistância de cada satélite. É precisamente neste ponto que o `generateClientRTCM()` se diferencia do `generateRTCM()`, ao carregar as coordenadas do cliente em questão na determinação do IPP, modificando assim a correcção ionosfera calculada analiticamente e, do mesmo modo, modificando também o valor final da correcção a aplicar à pseudodistância de cada satélite da constelação. Deste modo, cada cliente `SiRF Interface` ligado ao `EGNOS VRS Server` vai receber correcções personalizadas, calculadas com base na sua posição. Posteriormente, é invocado o método `makeClientRtcmMessage()` (em tudo semelhante ao método `makeRtcmMessage()`) que, com base nestas correcções optimizadas, gera a MT1 RTCM SC-104 optimizada para o cliente em questão. As mensagens MT1 RTCM SC-104 geradas segundo o conceito VRS contém os seguintes parâmetros: (i) PRC_{vrs} - a correcção da pseudodistância; (ii) RRC_{vrs} - a correcção *range-rate*; e (iii) MZC_{vrs} - o parâmetro *modified z-count* (MZC). O MZC é determinado através da Equação (5.50).

Equação 5.50: *Modified Z-Count*

$$MZC = T_{gs} \bmod 3600 \quad (s) \quad (5.50)$$

onde T_{gs} é o *GPS Software Time* contido na mensagem das correcções do EGNOS.

Excerto de Código 5.18 Método `generateClientRTCM()` da classe `SISNetClient`

```

1   public Vector generateClientRTCM(Rover rov){
2       Vector clientRTCMData = new Vector();
3       for (int i = 0; i < gps_const.dimSats; i++) {
4           if (gps_const.satellite[i].ephemeris.svid != 0
5               && gps_const.satellite[i].ephemeris.toc_str != null) {
6               if (gps_const.satellite[i].ephemeris.iode ==
7                   gps_const.satellite[i].longtermcorrection.IOPD
8                   && gps_const.satellite[i].ephemeris.iodec ==
9                       gps_const.satellite[i].longtermcorrection.IOPD) {
10                  gps_const.satellite[i].longtermcorrection.correct(
11                      ans.gps_time - (ans.gps_day * 60 * 60 * 24));
12              }
13              else {
14                  gps_const.satellite[i].longtermcorrection.corrAvailable = false;
15              }
16              gps_const.satellite[i].Ipp.setCords(gps_const.satellite[i],
17                  rov, ans.gps_time, (ans.gps_day * 60 * 60 * 24));
18              gps_const.satellite[i].Ipp.fillGrid();
19              gps_const.satellite[i].Ipp.resolveShape(gps_const.satellite[i].Ipp.Grid);
20              gps_const.satellite[i].Ipp.interpolate(ans.gps_time);
21              gps_const.satellite[i].Tropo.Calculate(gps_const.satellite[i], rov);
22              if (gps_const.satellite[i].ephemeris.iode ==
23                  gps_const.satellite[i].longtermcorrection.IOPD
24                  && gps_const.satellite[i].ephemeris.iodec ==
25                      gps_const.satellite[i].longtermcorrection.IOPD) {
26                  gps_const.satellite[i].longtermcorrection.correct(
27                      ans.gps_time - (ans.gps_day * 60 * 60 * 24));
28              }
29              else {
30                  gps_const.satellite[i].longtermcorrection.corrAvailable = false;
31              }
32          }
33      }
34      ...
35      makeClientRtcmMessage(1, clientRTCMData);
36      return clientRTCMData;
37  }

```

Este processo faz com que o módulo **EGNOS VRS Server** funcione como uma estação de referência virtual, ao simular de forma sistemática a existência de uma estação de referência coincidente com a posição do receptor cliente que gera correções de forma otimizada e independente da distância entre o receptor e o local físico onde está implementada a estação. A Figura (5.15) ilustra o atendimento de clientes **SiRF Interface** por parte do **EGNOS VRS Server**, nomeadamente a troca de mensagens entre entidades.

5.2.3 Serviço IDGPS

Este módulo é responsável pela implementação do conceito IDGPS. Recebe dos módulos **SiRF Interface** as correções diferenciais geradas pelo módulo **EGNOS VRS Server** e os dados brutos necessários à determinação da posição de cada cliente móvel e determina a sua posição corrigida. Este módulo adopta igualmente uma arquitectura cliente-servidor, onde o **IDGPS Server** é o servidor e os diversos módulos **SiRF Interface** ligados ao sistema de seguimento de plataformas móveis são os seus clientes. O **IDGPS Server** realiza as seguintes funções:

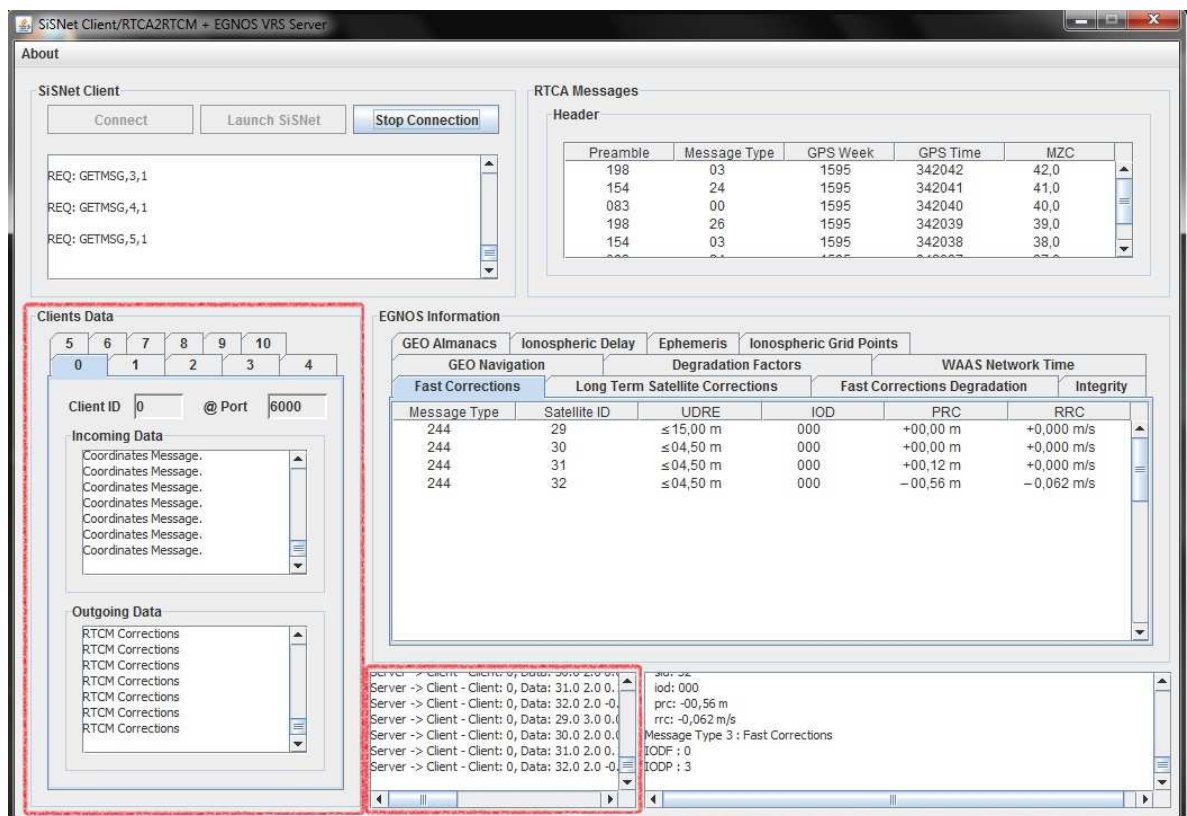


Figura 5.15: EGNOS VRS Server: atendimento dos clientes SiRF Interface

1. Criação de um servidor capaz de atender, em simultâneo, um máximo de 10 clientes³ SiRF Interface;
2. Implementação do conceito de IDGPS através da determinação para cada cliente de uma solução de posicionamento mais precisa com base nos dados recebidos (efemérides dos satélites visíveis, medidas brutas, coordenadas e correções RTCM SC-104 geradas pelo módulo EGNOS VRS Server);
3. Armazenamento das posições resultantes dos clientes SiRF Interface numa base de dados.

O desenvolvimento deste módulo passou, não só, pela utilização do ambiente de desenvolvimento de aplicações NetBeans IDE 6.9, mas também, pela utilização do MySQL Workbench 5.2 CE, uma ferramenta fundamental à implementação, interação e modelação de bases de dados do MySQL.

³Este número de clientes permitiu realizar a prova de conceito, não constituindo uma limitação do sistema.

5.2.3.1 Aquisição e Tratamento de Dados

A execução deste módulo é feita pela classe principal `IDGPServer`, cujo método construtor cria o ambiente gráfico do módulo e o `ServerSocket` que escuta a chegada de clientes ao porto 5060. O método `execute()` verifica, periodicamente, a chegada de clientes e, caso surja algum cliente, cria um objecto `HandleClient()` para representar o seu atendimento. O extracto de código (5.19) mostra estes passos.

Excerto de Código 5.19 Método construtor da classe principal `IDGPServer` e método `execute()` que escuta a chegada de clientes `SiRF Interface`

```

1  public class IDGPServer extends javax.swing.JFrame {
2      private final int dim_clients = 10;
3      private ServerSocket socket_server = null;
4      private Vector clients = null;
5      private int port = 5060;
6      ...
7      public IDGPServer() {
8          ...
9          clients = new Vector();
10         ...
11         try {
12             socket_server = new ServerSocket(port, dim_clients);
13             this.display("IDGPS Server Initiated!");
14             this.display("Listening to clients on port "+port+".");
15         }
16         catch (IOException e) {
17             e.printStackTrace();
18             this.display("Failed to initiate IDGPS Server!");
19         }
20     }
21     public void execute() {
22         try {
23             while(true) {
24                 if(clients.size() < dim_clients) {
25                     clients.addElement(new HandleClient(socket_server.accept(), this));
26                     ((HandleClient)clients.lastElement()).start();
27                 }
28                 //Quando atingir o número máximo de clientes
29                 else if(clients.size() == dim_clients) {
30                     this.display("Maximum number of clients exceeded!");
31                     Thread.sleep(1000);
32                     this.display("Checking if any client has disconnected itself...");
33                 }
34             }
35         }
36         catch (Exception e) {
37             e.printStackTrace();
38             System.exit(1);
39         }
40     }
41     ...
42 }

```

Cada objecto `HandleClient` é uma *thread* dedicada ao processamento das mensagens de um dado cliente `SiRF Interface`. O processamento é feito pelo método `processMsg()` da seguinte forma:

- Quando um cliente `SiRF Interface` se liga ao `IDGPS Server`, envia uma mensagem de identificação contendo a identificação atribuída ao móvel pelo `EGNOS VRS Server`;

- Após a confirmação da identificação, o **IDGPS Server** fica a aguardar a chegada periódica das mensagens contendo os dados necessários à determinação precisa da posição: parâmetros do modelo ionosférico (representada pelo método `processMsgIono()`), posição do receptor cliente (representada pelo método `processMsgPos()`), medidas brutas (representada pelo método `processMsgRaw()`), efemérides dos satélites visíveis pelo receptor cliente (representada por `processMsgEph()`) e correcções em formato RTCM SC-104 geradas pelo **EGNOS VRS Server** (representadas por `processMsgRTCM()`).

De notar que o método `processMsg()` recebe como argumento uma mensagem do protocolo `ProtocolIDGPS` - ver extracto de código (5.20).

Excerto de Código 5.20 IDGPS Server: processamento das mensagens dos clientes **SiRF Interface**

```

1  public void run() {
2      try {
3          while(!done) {
4              processMsg(ProtocolIDGPS.receive(in_data));
5          }
6          Thread.sleep(1500);
7          conn.close();
8          this.server.display("Socket closed!");
9      }
10     catch(Exception e) {
11         e.printStackTrace();
12     }
13 }
14 public void processMsg(ProtocolIDGPS msg) throws Exception {
15     if(msg instanceof MsgIdt)
16         processMsgIdt((MsgIdt)msg);
17     else if(msg instanceof MsgIono)
18         processMsgIono((MsgIono)msg);
19     else if(msg instanceof MsgPos)
20         processMsgPos((MsgPos)msg);
21     else if(msg instanceof MsgRaw)
22         processMsgRaw((MsgRaw)msg);
23     else if(msg instanceof MsgEph)
24         processMsgEph((MsgEph)msg);
25     else if(msg instanceof MsgRTCM)
26         processMsgRTCM((MsgRTCM)msg);
27     else if(msg instanceof MsgEnd)
28         processMsgEnd((MsgEnd)msg);
29 }

```

Os dados recebidos dos clientes **SiRF Interface** são armazenados em objectos `clientPanel`. Esta classe não só tem estruturas de dados para conter todos os dados recebidos, mas também cria o ambiente gráfico utilizado para apresentar a informação relativa a cada cliente do **IDGPS Server**. A Figura (5.16) mostra a recepção de dados provenientes dos clientes **SiRF Interface** assim como os resultados obtidos.

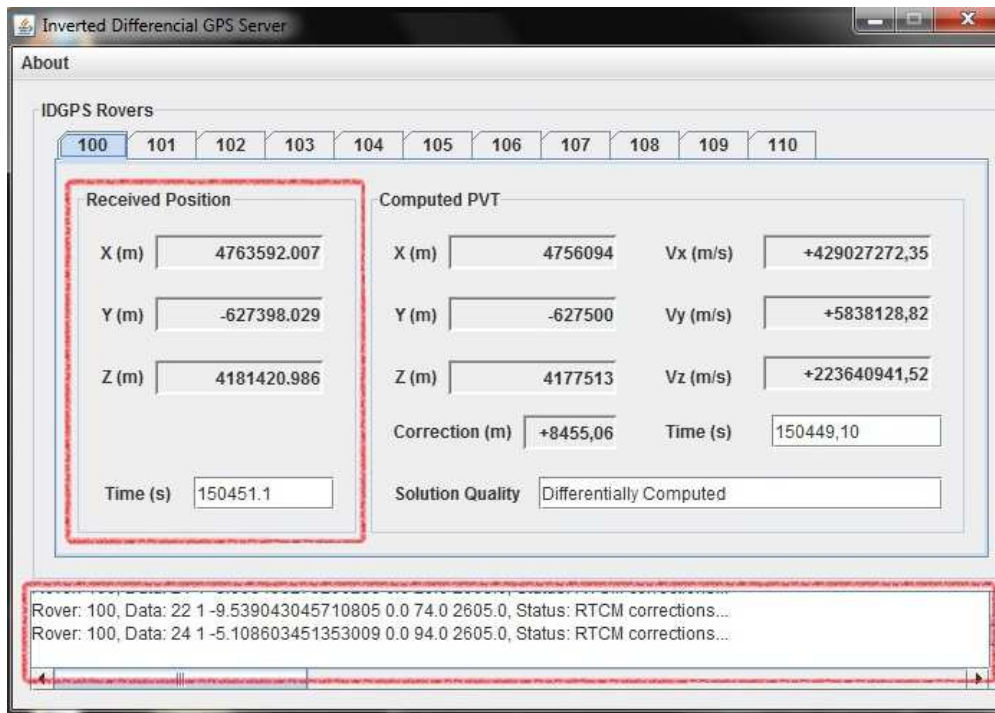


Figura 5.16: IDGPS Server: atendimento dos clientes SiRF Interface

5.2.3.2 Posicionamento em Modo Diferencial

Uma vez na posse de todos os dados necessários dos clientes SiRF Interface, passou-se à implementação do conceito de IDGPS, *i.e.*, à determinação num nó central das posições precisas de um conjunto de móveis a partir das suas respectivas medidas brutas e correções diferenciais. Este algoritmo corresponde ao algoritmo de posicionamento em modo diferencial (PosDif).

O processo de determinação da solução PVT diferencial de um receptor GNSS passa pelos seguintes pontos:

- Determinação da posição dos satélites visíveis;
- Aplicação das correções diferenciais às pseudodistâncias observadas;
- Determinação da posição do receptor com base na posição dos satélites visíveis e das pseudodistâncias corrigidas.

Como pode ver-se, este processo é muito semelhante ao descrito em (5.2.1.2), onde são descritos os algoritmos de posicionamento dos receptores GNSS ligados ao módulo SiRF Interface.

As posições dos satélites visíveis são determinadas a partir das suas efemérides segundo o algoritmo descrito em (5.2.1.2). O mesmo acontece para a determinação da posição do receptor. Somente a determinação das correcções das pseudodistâncias dos satélites é distinta da referida em (5.2.1.2) porque, no caso do IDGPS *Server*, podem surgir duas situações:

1. Se o cliente em questão enviou correcções RTCM SC-104 para o IDGPS *Server* e o IDO da mensagem RTCM SC-104 é coincidente com o IODE das últimas efemérides recebidas, então é possível aplicar as correcções recebidas às pseudodistâncias dos satélites visíveis. Em primeiro lugar, de acordo com [12], determina-se a correcção diferencial de cada satélite através da Equação (5.51) para o instante t .

Equação 5.51: Correcção diferencial da pseudodistância

$$PRC_{dif}(t) = PRC_{vrs}(t_0) + RRC_{vrs}(t - t_0) \quad (m) \quad (5.51)$$

onde $PRC_{dif}(t)$ é a correcção diferencial da pseudodistância, $PRC_{vrs}(t_0)$ e RRC_{vrs} são a correcção da pseudodistância e a correcção *range-rate* contidas na MT1 RTCM SC-104, t é o tempo actual e t_0 é o *modified z-count* contido na MT1 RTCM SC-104. Em segundo lugar, determina-se a pseudodistância ajustada PR_{adj} para o instante t de cada satélite através da (5.52).

Equação 5.52: Pseudodistância ajustada

$$PR_{adj}(t) = PR_{obs}(t) + dTclk(t) * c \quad (m) \quad (5.52)$$

onde $PR_{obs}(t)$ é a pseudodistância observada no instante t da Equação (5.1) e $dTclk$ é o desvio de relógio no instante t obtido através da Equação (5.27). Em terceiro e último lugar, determina-se a pseudodistância de cada satélite no instante t corrigida em modo diferencial $PR_{dif}(t)$ através da Equação (5.53).

Equação 5.53: Pseudodistância corrigida em modo diferencial

$$PR_{dif}(t) = PR_{adj}(t) + PRC_{dif}(t) \quad (m) \quad (5.53)$$

em que $PR_{adj}(t)$ é a pseudodistância ajustada da Equação (5.52) e $PRC_{dif}(t)$ é a correcção diferencial determinada pela Equação (5.51).

2. Se o cliente em questão não enviou correcções RTCM SC-104 para o IDGPS *Server* ou, se enviou, mas o IDO da mensagem RTCM SC-104 não é coincidente com o IODE das últimas efemérides recebidas, então não é possível

aplicar as correções recebidas às pseudodistâncias dos satélites visíveis. Quando isto acontece, é aplicado o algoritmo descrito em (5.2.1.2).

Este processo é executado pela *thread* `ComputePVT`, que invoca, a cada 2 s, o método `updatePVT()` da classe `clientPanel`. Este método `updatePVT()` implementa o algoritmo de posicionamento diferencial `PosDif` que é utilizado para determinar a posição em modo diferencial de cada cliente `SiRF Interface` ligado ao `IDGPS Server`. Os dados recebidos de cada cliente são armazenados em estruturas `RoverData` e `RoverSatData` que são, posteriormente, acedidas pelo `updatePVT()` para determinar a posição diferencial. A Figura (5.17) mostra os resultados na interface gráfica do módulo `IDGPS Server`.

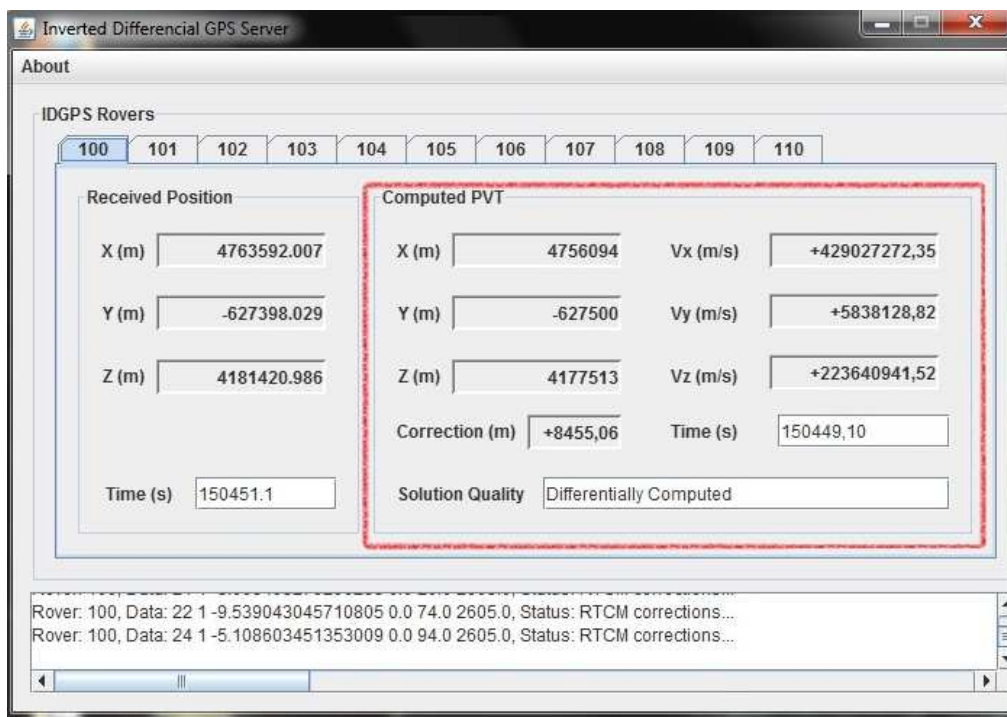


Figura 5.17: IDGPS Server: resultados do método `updatePVT()`

5.2.3.3 Armazenamento de Dados

O armazenamento dos dados no módulo `IDGPS Server` é feito numa base de dados MySQL, desenvolvida através da ferramenta de modelação e concepção de bases de dados MySQL Workbench CE.

Uma vez que se pretende desenvolver um sistema de seguimento de plataformas móveis a aplicar ao contexto dos transportes, foi desenvolvida uma estrutura que permita acomodar toda a informação aplicável a este cenário:

- Uma tabela para acomodar os dados dos *rovers* ligados ao sistema;
- Uma tabela para acomodar a informação relativa aos condutores dos *rovers*;
- Uma tabela para acomodar a informação relativa ao operador de transportes que utilize este sistema de seguimento de plataformas;
- Uma tabela com a informação relativa aos veículos pertencentes à frota;
- Três tabelas para relacionar os veículos com os *rovers*, os condutores com o operador de transportes e os veículos com o operador de transportes.

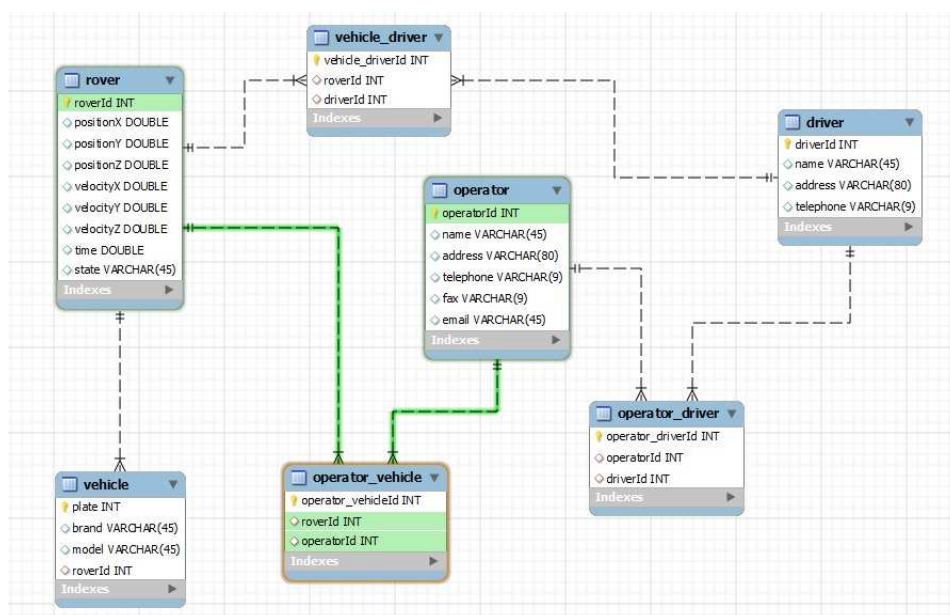


Figura 5.18: Modelo da base de dados trackingDB

Esta estrutura de dados, denominada **trackingDB**, encontra-se ilustrada na Figura (5.18) e é constituída pelas seguintes tabelas e campos:

Tabela rover - É a tabela utilizada para acomodar informação específica acerca dos móveis ligados ao sistema de seguimento de plataformas. Contém os seguintes campos: identificação do *rover*, componentes *X*, *Y* e *Z* da posição do *rover*, componentes *X*, *Y* e *Z* da velocidade do *rover*, instante temporal da medida e estado do *rover*.

Tabela driver - É a tabela utilizada para acomodar informação acerca dos condutores dos móveis ligados ao sistema. Contém campos com o identificador do condutor, o seu nome, a sua morada e contacto telefónico.

Tabela operator - É a tabela utilizada para acomodar informação acerca dos operadores de transportes que utilizam o sistema de seguimento de plataformas desenvolvido. Contém um identificador do operador de transportes, nome, morada, contacto telefónico, contacto de correio electrónico, fax e identificador dos condutores empregados pelo operador.

Tabela vehicle - É a tabela utilizada para acomodar informação dos veículos da frota. Contém informação da matrícula, marca e modelo do veículo bem como do identificador do *rover* que lhe está associado.

Tabela operator_vehicle - É a tabela que relaciona que veículos estão associados a cada operador.

Tabela operator_driver - É a tabela que relaciona que condutores estão associados a cada operador.

Tabela vehicle_driver - É a tabela que relaciona os condutores com os *rovers*, ou seja, indica que condutores estão associados a cada *rover*.

O acesso à base de dados no módulo **IDGPS Server** é feito através de um *java bean* designado **TrackingBean**. Um *bean* é uma estrutura de dados dinâmica, capaz de interagir com bases de dados. Apresenta um conjunto de métodos, vulgarmente designados por *setters* e *getters*, para definição e acesso aos dados armazenados na estrutura do *bean*, e um segundo conjunto de métodos para passar da estrutura do *bean* para a base de dados e da base de dados para a estrutura do *bean*. No caso deste módulo, apenas se pretende preencher a base de dados com a informação dos clientes **SiRF Interface** e não tanto consultar essa informação. Assim, utilizam-se para armazenar a informação na estrutura de dados do *bean* os diversos *setters* do *bean* - ver extracto de código (5.21).

A invocação do *bean* é feita pelo método `toDataBase()` da classe `clientPanel`, que recebe como argumento a posição, velocidade e instante temporal do receptor **SiRF Interface** ligado ao **IDGPS Server**. Este método trata de invocar os *setters* necessários assim como o método `updateDBRecord()`. O `updateDBRecord()` estabelece a ligação à base de dados, cria e submete o comando SQL que armazena nas tabelas o conteúdo dos *setters*. Os extractos de código (5.22) e (5.23) apresentam os métodos `toDataBase()` e `updateDBRecord()`.

De notar que, quando um cliente **SiRF Interface** se liga ao **IDGPS Server**, não só é iniciada a passagem de informação de posição, velocidade e instante temporal para a base de dados `trackingDB`, mas também o estado do *rover* passa a "ONLINE" através *setter* `setState()`. Do mesmo modo, quando um cliente fecha a ligação ao **IDGPS Server**, o *setter* `setState()` coloca o *rover* no estado "OFFLINE". Este pormenor garante que só podem ser seguidos *rovers* ligados. No

Excerto de Código 5.21 *Setters e getters* do *bean* utilizado para interagir com a base de dados *TrackingBean*

```

1  public class TrackingBean {
2      ...
3      public void setRoverID(int rid) {RoverID = rid;}
4      public int getRoverID() {return RoverID;}
5      public void setRoverXp(double xpos) {RoverXp = xpos;}
6      public double getRoverXp() {return RoverXp;}
7      public void setRoverYp(double ypos) {RoverYp = ypos;}
8      public double getRoverYp() {return RoverYp;}
9      public void setRoverZp(double zpos) {RoverZp = zpos;}
10     public double getRoverZp() {return RoverZp;}
11     public void setRoverXv(double xvel) {RoverXv = xvel;}
12     public double getRoverXv() {return RoverXv;}
13     public void setRoverYv(double yvel) {RoverYv = yvel;}
14     public double getRoverYv() {return RoverYv;}
15     public void setRoverZv(double zvel) {RoverZv = zvel;}
16     public double getRoverZv() {return RoverZv;}
17     public void setRoverT(double time) {RoverT = time;}
18     public double getRoverT() {return RoverT;}
19     public void setState(String st) {State = st;}
20     public String getState() {return State;}
21     ...
22 }

```

Excerto de Código 5.22 Método *toDataBase()* da classe *clientPanel*, que invoca os *setters* do *TrackingBean*

```

1  public void toDataBase(int xp, int yp, int zp, double xv, double
2      yv, double zv, double t) {
3      if(!connected) {
4          bean = new TrackingBean();
5          bean.setRoverID(this.id);
6          bean.updateDBRecord();
7          connected = true;
8      }
9      bean.setRoverXp(xp);
10     bean.setRoverYp(yp);
11     bean.setRoverZp(zp);
12     bean.setRoverXv(xv);
13     bean.setRoverYv(yv);
14     bean.setRoverZv(zv);
15     bean.setRoverT(t);
16     bean.setState("ONLINE");
17     bean.updateDBRecord();
18 }

```

entanto, a informação de um *rover* desligado não é apagada da base de dados, sendo sempre mantida a última posição conhecida.

5.2.4 Aplicação Web Tracking of Mobile Platforms

Desenvolvidos todos os módulos de *software* que constituem o sistema de seguimento de plataformas móveis, passou-se à implementação da aplicação *Web* que permite o acesso à informação dos móveis através da Internet. A adoção de uma aplicação *Web*, ao invés de uma aplicação *stand-alone*, apresenta diversas vantagens, sendo a mais relevante a garantia da portabilidade entre *desktops*, *laptops* e *smartphones*. Assim, esta aplicação deve:

Excerto de Código 5.23 Método `updateDBRecord()` da classe `TrackingBean`, que estabelece a ligação com a base de dados e preenche as tabelas

```

1  public void updateDBRecord() {
2      try {
3          Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
4          Connection conn = DriverManager.getConnection("jdbc:odbc:trackingDB", "root","root");
5          Statement st = conn.createStatement();
6          String query = "UPDATE ROVER SET positionX='"+RoverXp+"', positionY='"+RoverYp+"',
7              positionZ='"+RoverZp+"', velocityX='"+RoverXv+"', velocityY='"+RoverYv+"',
8              velocityZ='"+RoverZv+"', time='"+RoverT+"', state='"+State+"
9              WHERE roverId='"+RoverID+"'";
10         System.out.println(query);
11         st.executeUpdate(query);
12         st.close();
13         conn.close();
14         System.out.println("Rover data successfully updated to DB!");
15     }
16     catch(Exception e){
17         System.out.println("Error saving rover data to DB!" +e);
18     }
19 }

```

- Permitir o acesso, através da Internet, a uma página *Web* com a informação do sistema de seguimento de plataformas móveis;
- Permitir ao utilizador seleccionar quais os móveis a seguir;
- Obter da base de dados a informação necessária ao seguimento dos móveis seleccionados;
- Disponibilizar a informação de seguimento sobre um mapa.

A aplicação *Web*, definida através da classe `MapServlet`, é um *Servlet* que processa os pedidos submetidos através de uma página *Web* e envia respostas dinâmicas aos clientes. Os pedidos são feitos através do método `doGet()`, que processa pedidos do tipo HTTP GET e recebe como argumento os objectos `HttpServletRequest` e `HttpServletResponse` (pedido efectuado pelo cliente e resposta a enviar ao cliente, respectivamente). O código HTML de resposta é gerado neste método como mostra o extracto de código (5.24) e enviado para o cliente através de um objecto `PrintWriter`.

Como já foi referido anteriormente, o `EGNOS VRS Server` e o `IDGPS Server` estão actualmente configurados para atender, no máximo, 10 clientes em simultâneo. Isto implica que o sistema de seguimento de plataformas permite seguir uma frota de 10 *rovers*. Assim, a página *Web* apresenta ao utilizador um conjunto de botões (um por *rover*), cabendo ao utilizador seleccionar o *rover* que pretende seguir. Estes botões são monitorados através do objecto `request`. Quando um botão é pressionado, é invocado o método `getRoverTracking()`. Este processo encontra-se descrito no extracto de código (5.25).

O método `getRoverTracking()` é o responsável pela interacção com a base de dados. Constrói uma *query* em função do botão pressionado que é utilizada para

Excerto de Código 5.24 *Servlet* MapServlet e o seu método doGet() que processa os pedidos dos clientes

```

1  public class MapServlet extends HttpServlet {
2      public void doPost(HttpServletRequest request,
3          HttpServletResponse response) throws IOException, ServletException {
4          doGet(request, response);
5      }
6      public void doGet(HttpServletRequest request,
7          HttpServletResponse response) throws IOException, ServletException {
8          response.setContentType("text/html");
9          PrintWriter out = response.getWriter();
10         out.println("<html>");
11         out.println("<head><title>Tracking of Mobile Platforms Web Application</title>"
12             + "<meta name=\"viewport\" content=\"initial-scale=1.0, user-scalable=no\" />"
13             + "<link rel=\"stylesheet\" type=\"text/css\" "
14             + "href=\"http://www.dee.isep.ipp.pt/~mbm/css/bene.css\">"
15             + "<link rel=\"icon\" type=\"image/jpg\"
16             + "href=\"http://www.dee.isep.ipp.pt/~mbm/images/isep_logo.jpg\">"
17             + "<link rel=\"SHORTCUT ICON\" "
18             + "href=\"http://www.dee.isep.ipp.pt/~mbm/images/isep_logo.jpg\">");
19         ...
20         out.println("<td class=\"fundo_branco\">"
21             + "This Web application was developed to track a limited number "
22             + "of mobile platforms, called rovers. By default up to 10 rovers can be "
23             + "simultaneously connected. A rover can be tracked as long as it is connected. "
24             + "To track a specific rover, please click on the corresponding button.</td>");
25         ...
26     }
27 }

```

Excerto de Código 5.25 Escuta dos botões que representam os *rovers* feita pelo objecto request

```

1  public void doGet(HttpServletRequest request,
2      HttpServletResponse response) throws IOException, ServletException {
3      ...
4      out.println("<input type=\"submit\" name=\"op\" value=\"Track Rover 100\"><br><br>");
5      out.println("<input type=\"submit\" name=\"op\" value=\"Track Rover 101\"><br><br>");
6      out.println("<input type=\"submit\" name=\"op\" value=\"Track Rover 102\"><br><br>");
7      out.println("<input type=\"submit\" name=\"op\" value=\"Track Rover 103\"><br><br>");
8      out.println("<input type=\"submit\" name=\"op\" value=\"Track Rover 104\"><br><br>");
9      out.println("<input type=\"submit\" name=\"op\" value=\"Track Rover 105\"><br><br>");
10     out.println("<input type=\"submit\" name=\"op\" value=\"Track Rover 106\"><br><br>");
11     out.println("<input type=\"submit\" name=\"op\" value=\"Track Rover 107\"><br><br>");
12     out.println("<input type=\"submit\" name=\"op\" value=\"Track Rover 108\"><br><br>");
13     out.println("<input type=\"submit\" name=\"op\" value=\"Track Rover 109\"><br><br>");
14     out.println("<input type=\"submit\" name=\"op\" value=\"Track Rover 110\"><br><br>");
15     ...
16     if(request.getParameter("op") != null) {
17         ...
18         answer = getRoverTracking(request);
19         ...
20     }
21     ...
22 }

```

questionar a base de dados e retornar os registos relativos ao *rover* seleccionado pelo cliente. Neste ponto, podem surgir três situações diferentes:

1. Não existem registos para o *rover* seleccionado - é retornada uma mensagem "No data stored in data base for the specified rover!";
2. Existem registos para o *rover* seleccionado mas ele encontra-se "OFFLINE" - é retornada uma mensagem "Rover is offline and cannot be tracked!";
3. Existem registos para o *rover* seleccionado e ele encontra-se "ONLINE" - é retornada uma *string* com as coordenadas ECEF do *rover* extraídas da base de dados convertidas para coordenadas geográficas.

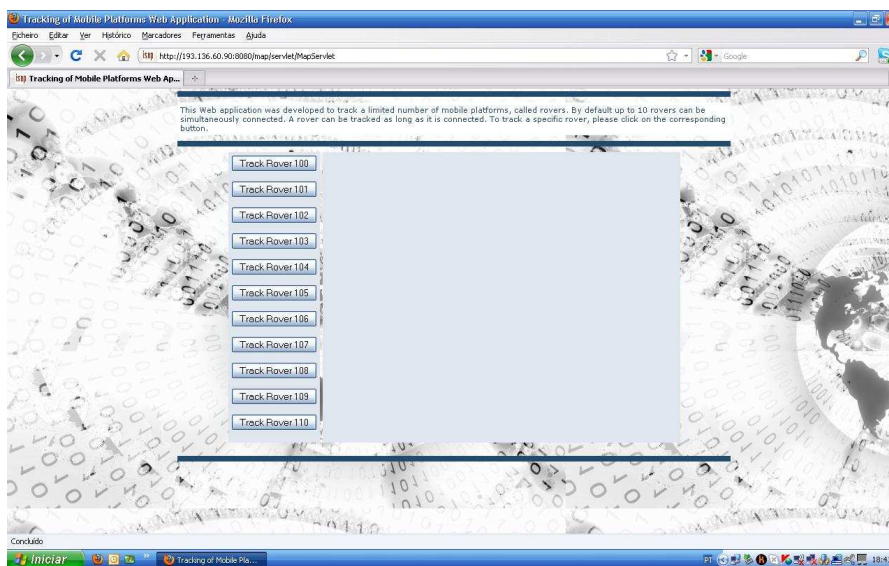
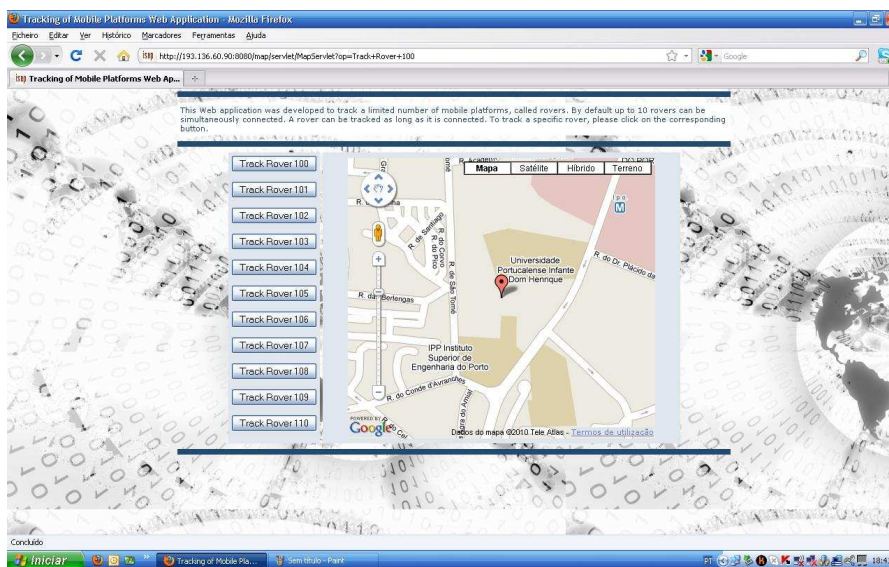
Com estas coordenadas, é então criado um mapa onde se visualiza a posição do móvel. Isto é feito através da API do Google Maps [27] que disponibiliza um serviço de geração de mapas centrados em coordenadas específicas. A chamada à API do Google Maps é feita através de um *script* de JavaScript, carregado no cabeçalho da página *Web* - ver excerto de código (5.26). A geração do mapa é feita pela função `initialize()`, que também se encontra descrita no extracto de código (5.26). Assim, sempre que são pedidos dados de um *rover* diferente ou a página é actualizada, o mapa também é actualizado.

Excerto de Código 5.26 *Script* de JavaScript que faz a invocação da API do Google Maps

```

1  public void doGet(HttpServletRequest request, HttpServletResponse response)
2      throws IOException, ServletException {
3      ...
4      out.println("<script type=\"text/javascript\">
5          + \"src=\"http://maps.google.com/maps/api/js?sensor=true\">\"
6          + \"</script>\";
7      out.println("<script type=\"text/javascript\">\n\"
8          + \"function initialize() {\n\"
9          + \"var latlng = new google.maps.LatLng(
10             + \"        document.getElementById(\"latitude\").value,
11             + \"        document.getElementById(\"longitude\").value);\n\"
12          + \"var myOptions = {\n\"
13          + \"    zoom: 16,\n\"
14          + \"    center: latlng,\n\"
15          + \"    mapTypeId: google.maps.MapTypeId.ROADMAP,\n\"
16          + \"    streetViewControl: true};\n\"
17          + \"var map = new google.maps.Map(document.getElementById(\"map_canvas\"),myOptions);\n\"
18          + \"var marker = new google.maps.Marker({position: latlng, map: map, title: \"Rover\"}); \n\"
19          + \"google.maps.event.addDomListener(window, 'load', initialize);\n\"
20          + \"}\n\"
21          + \"</script>\n\"
22          + \"</head>\";
23      out.println("<body background=\"/map/images/digital_world800_600.jpg\"onload=\"initialize()\">\";
24  }
```

Por fim, o *servlet* foi alojado no servidor de aplicações Apache Tomcat, ficando disponível no porto 8080 do endereço 193.136.60.90. As Figuras (5.19) e (5.20) mostram a página *Web* gerada pela aplicação *Web Tracking of Mobile Platforms*

Figura 5.19: Página Web da aplicação *Web Tracking of Mobile Platforms*Figura 5.20: Seguimento de um *rover*

assim como o mapa da Google centrado nas coordenadas extraídas da base de dados trackingDB.

5.3 Conclusão

Neste capítulo, dedicado ao desenvolvimento do sistema de seguimento de plataformas móveis, detalhou-se a arquitectura do sistema, os seus módulos constituintes e as funcionalidades que realizam.

Foram descritos de forma pormenorizada os algoritmos de posicionamento em modo absoluto e em modo diferencial desenvolvidos assim como a geração de correcções diferenciais segundo a técnica VRS a partir das correcções de área alargada do EGNOS e da sua aplicação segundo o conceito IDGPS.

Capítulo 6

Resultados dos Testes e Depuração do Código

Neste capítulo verifica-se o correcto funcionamento da aplicação e testam-se as diversas funcionalidades.

6.1 Resultados dos Testes

Ao longo do desenvolvimento dos diversos módulos constituintes do sistema de seguimento de plataformas móveis foram feitos diversos testes das funcionalidades implementadas em cada módulo.

Durante os testes efectuados aos módulos desenvolvidos, foram verificados diversos resultados:

- A aquisição, decodificação e tratamento dos dados emitidos pelo receptor GNSS Holux GM-210 em protocolo SiRF feita pelo módulo **SiRF Interface** foi verificada com sucesso. Esta afirmação pode ser comprovada com base na comparação dos diversos dados obtidos (nomeadamente medidas brutas e efemérides) com os dados de navegação e medidas observáveis contidas nos ficheiros RINEX disponibilizados por <http://navstar.idt.ipp.pt/>.
- A determinação analítica da solução de posicionamento do receptor através do algoritmo **PosAbs** no módulo **SiRF Interface** foi bem sucedida. Os resultados encontram-se validados e documentados em diversos ficheiros de registo retirados ao longo dos diversos testes efectuados, cujos resultados se encontram nas Figuras (6.1) e (6.2). A solução calculada e a solução

produzida pelo *firmware* do receptor contida na mensagem SiRF 4 diferem de poucas dezenas de metros, como ilustrado nas Figuras (6.3), (6.4) e (6.5).

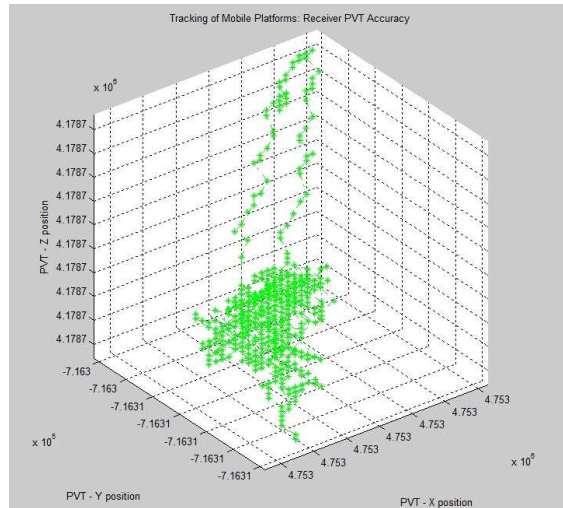


Figura 6.1: Solução PVT do receptor Holux GM-210

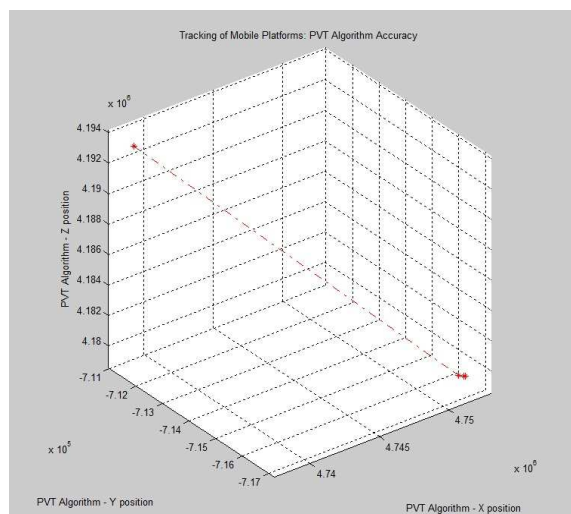


Figura 6.2: Solução PVT do PosAbs

- O módulo **EGNOS VRS Server** realiza com sucesso todas as suas funcionalidades previstas. Recebe periodicamente as diversas mensagens RTCA DO-229 do SISNeT assim como as efemérides da constelação. Com base nessas mensagens, converte-as correctamente em mensagens RTCM SC-104, obedecendo na íntegra às recomendações da especificação do protocolo.

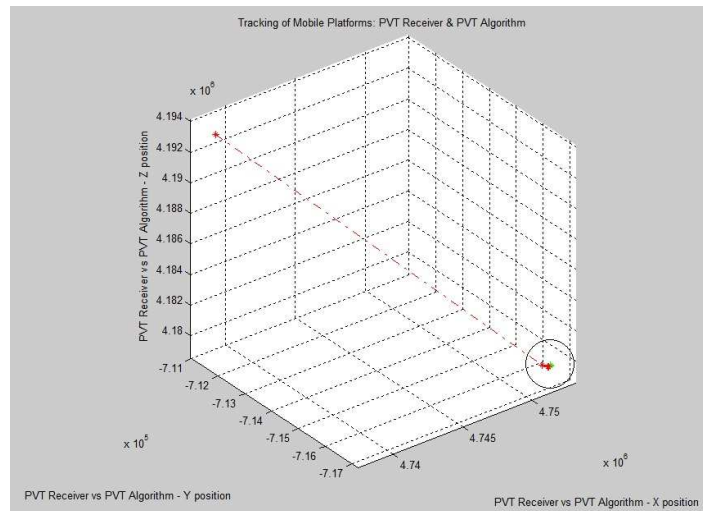


Figura 6.3: Solução PVT: receptor *vs* PosAbs

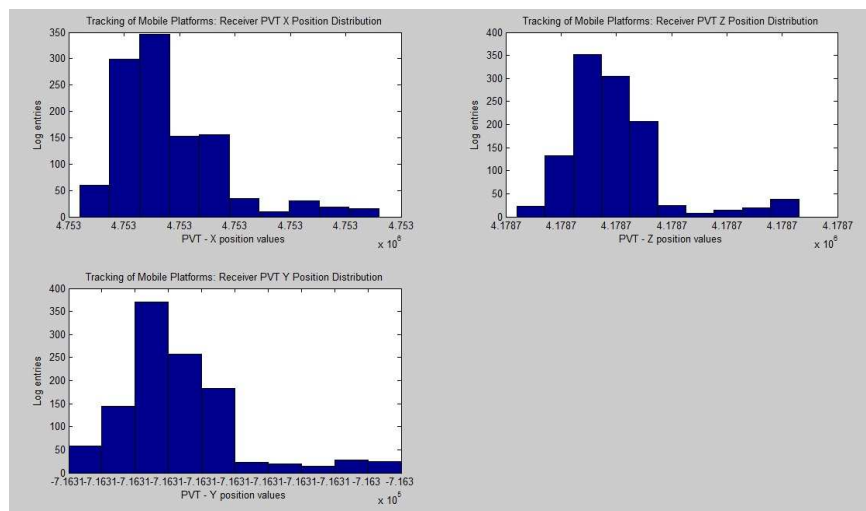


Figura 6.4: Solução PVT: distribuição dos valores do receptor

- O módulo **EGNOS VRS Server** estabelece a ligação ao **SISNeT** e recebe a informação do sistema **EGNOS**. No entanto, o **SISNeT** a certas horas do dia (às 15:00 e às 23:00) e durante cerca de 1 h, não retorna dados válidos, *i.e.*, os campos **IDO** de todas as mensagens têm valor 0. Esta situação foi analisada ao longo do período de testes, tendo-se concluído que se trata de períodos de manutenção do serviço **SISNeT**.
- As mensagens **RTCM SC-104** geradas para módulo **SiRF Interface** ligado ao **EGNOS VRS Server** correspondem às correcções a efectuar às pseudodis-

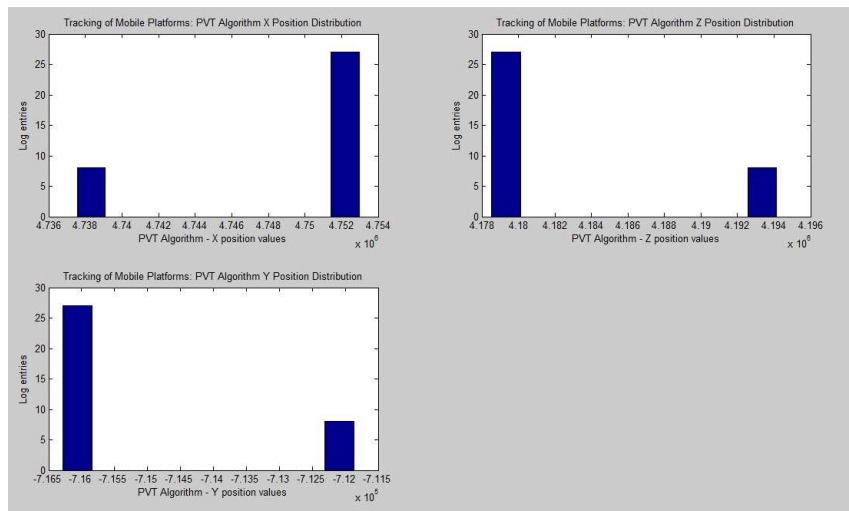


Figura 6.5: Solução PVT: distribuição dos valores do PosAbs

tâncias dos satélites calculadas com base na informação de área alargada obtida e na posição do receptor GNSS em questão. Isto foi verificado com base nos diversos testes realizados e através de ficheiros de registo, onde são armazenadas as mensagens RTCM SC-104 recebidas por cada **SiRF Interface**.

- O módulo **IDGPS Server** recebe, com sucesso, a informação dos vários receptores GNSS ligados aos módulos **SiRF Interface**. Apesar de esta ligação só ser possível quando estiver estabelecida uma ligação ao **EGNOS VRS Server**, as vantagens desta restrição superam largamente as desvantagens que lhe possam ser associadas: é garantida a uniformização na identificação dos receptores junto dos seus servidores assim como a existência do conjunto de dados necessários aos cálculos a efectuar por este módulo.
- A determinação analítica das soluções de posicionamento diferencial dos clientes do servidor **IDGPS Server** foi conseguida com sucesso após exaustiva depuração. Os resultados dos testes apontam para diferenças de posicionamento de poucos metros, quando comparadas a posição diferencial calculada analiticamente com a obtida da mensagem **SiRF 4** e enviada para este módulo em conjunto com as medidas brutas, efemérides, correcções RTCM SC-104 e parâmetros do modelo ionosférico. As Figuras (6.6) e (6.7) mostram os resultados obtidos do algoritmo **PosDif** durante um teste realizado, onde é evidente a coerência entre os valores PVT a serem enviados pelo cliente **SiRF Interface** e os determinados analiticamente.
- O armazenamento de informação na base de dados **trackingDB** pelo **IDGPS**

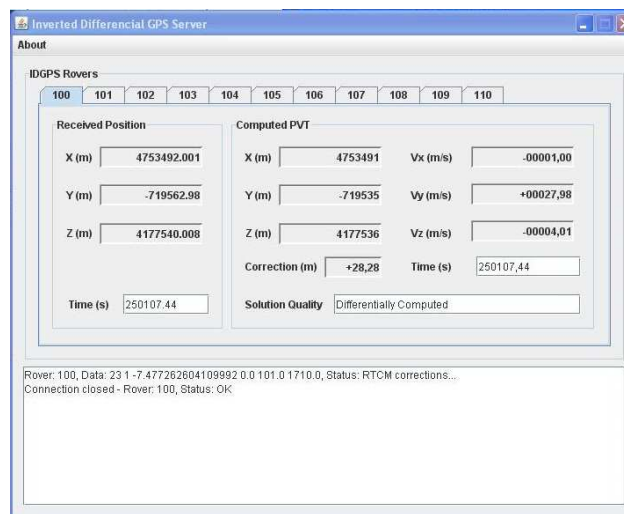


Figura 6.6: IDGPS Server em execução

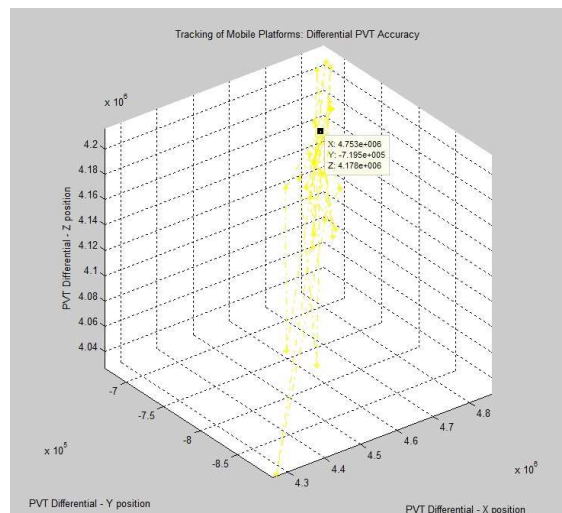


Figura 6.7: Solução PVT do PosDif

Server é realizado com sucesso. As posições dos receptores ligados aos SiRF Interface são armazenados nas tabelas com sucesso, não existindo quaisquer problemas de desempenho do sistema quando vários módulos escrevem na base de dados. De notar que esta afirmação é efectuada com base em testes efectuados com quatro clientes ligados ao sistema simultaneamente pois o Laboratório de Comunicações por Satélite do ISEP apenas dispõe de quatro receptores Holux GM-210 para teste.

- O acesso à informação das plataformas móveis através da aplicação Web

garante a interoperabilidade entre sistemas operativos, plataformas e dispositivos. Basta uma ligação à Internet e um navegador para se aceder ao sistema de seguimento de plataformas.

- A utilização da API do Google Maps permitiu disponibilizar uma interface adequada e amigável para o seguimento das plataformas. No entanto, a velocidade de carregamento dos mapas na página *Web* depende naturalmente da largura de faixa da ligação à Internet.

Com base nestes resultados, pode afirmar-se que o sistema de seguimento de plataformas móveis se encontra operacional e cumpre todas as funcionalidades propostas para cada módulo.

6.2 Depuração do Código

A depuração do código consistiu na identificação de problemas e na sua correcção, especialmente em relação aos algoritmos de determinação da posição nos modos absoluto e diferencial e às interfaces gráficas dos diversos módulos. Esta etapa foi uma das mais longas e importantes do trabalho, tendo a determinação da posição dos satélites e da posição em modo diferencial ocupado grande parte do tempo despendido nesta tarefa.

Para assegurar a escalabilidade e incrementar a usabilidade do sistema, seria ainda necessário realizar as seguintes tarefas:

- Analisar o comportamento do sistema face ao aumento do número de clientes ligados aos módulos servidores *EGNOS VRS Server* e *IDGPS Server* e, eventualmente, gerir de forma mais eficiente o fim e início das ligações.
- Enriquecer a interface com o utilizador gerada pela aplicação *Web*, incluindo algumas das funcionalidades descritas na API do Google Maps, tais como a marcação de pontos de referência ou *waypoints, etc.*
- Redesenhar a base de dados para acomodar mais informação, nomeadamente para comportar informação relevante para a gestão de frotas.

6.3 Conclusão

Neste capítulo descreveram-se os testes realizados e apresentaram-se os resultados obtidos. Referiu-se também a longa etapa de depuração do código efectuada, nomeadamente dos algoritmos de posicionamento absoluto e diferencial.

Capítulo 7

Conclusão e Desenvolvimentos Futuros

Neste capítulo realiza-se o balanço do trabalho realizado face aos objectivos iniciais e sugerem-se alguns eventuais desenvolvimentos futuros da aplicação.

7.1 Conclusão

O protótipo do sistema de seguimento de plataformas móveis foi desenvolvido com sucesso, podendo neste momento ser adoptado por qualquer operador de transportes detentor de uma frota de veículos de pequena dimensão.

A arquitectura concebida permite distribuir a carga computacional por diversas plataformas e vislumbrar uma escalabilidade suave do sistema. As tecnologias de suporte são independentes de plataforma e do tipo código aberto.

No seu desenvolvimento foram reutilizados e adaptados alguns módulos realizados por antigos alunos do ISEP, assim como foram desenvolvidos novos módulos capazes de lidar com a complexidade de um sistema desta envergadura. O sistema descrito nesta dissertação implementa todas as funcionalidades descritas nos objectivos iniciais:

- Recolhe os dados recebidos de uma frota de receptores GNSS utilizando o protocolo SiRF nomeadamente as medidas brutas e as efemérides;
- Determina analiticamente a posição dos receptores GNSS a partir dos dados da mensagem de navegação e das medidas brutas efectuadas pelos receptores;

- Adquire, através da Internet, as correcções de área alargada do EGNOS e converte-as do formato RTCA DO-229 para o formato RTCM SC-104 em tempo útil e de acordo com o conceito de estação de referência virtual;
- Envia para a frota de receptores GNSS as correcções diferenciais geradas no formato RTCM SC-104;
- Implementa o conceito de *Inverted Differential GNSS* ao aplicar num nó central as correcções diferenciais personalizadas às medidas brutas de cada receptor GNSS e determina as suas respectivas posições em modo diferencial;
- Armazena numa base de dados a informação da posição do conjunto de receptores;
- Disponibiliza a informação armazenada na base de dados numa interface do tipo aplicação *Web*, permitindo o seguimento das plataformas móveis, *i.e.*, da frota de receptores GNSS, através da Internet e sobre um mapa.

Num trabalho desta envergadura é habitual que surjam diversas dificuldades. A depuração dos algoritmos de posicionamento absoluto e diferencial são um exemplo destas dificuldades, porque exigiram de um grande investimento de tempo e esforço. No entanto, considera-se que todas as adversidades foram ultrapassadas com sucesso.

7.2 Desenvolvimentos Futuros

A continuação deste trabalho passa pelas seguintes vertentes:

- Substituição na aplicação *Web* do conjunto fixo de botões que representam os *rovers* por um controlo do tipo *pull-down* que, mediante uma consulta à base de dados, apenas mostra os móveis ligados ao sistema. Este melhoramento pode ser efectuado utilizando a tecnologia *Asynchronous JavaScript And XML* (AJAX);
- Permitir que o módulo *SiRF Interface*, através de uma opção no menu, aplique as correcções diferenciais obtidas do EGNOS *VRS Server* directamente ao receptor;
- Criação de módulos que interajam com outros receptores GNSS além do Holux, tais como o Garmin ou o u-Blox, que utilizam protocolos de comunicação proprietários;

- Aplicação deste sistema a interfaces mais reduzidas como PDA ou *smart-phones* que têm receptores GNSS incorporados;
- Depuração exaustiva de todos os módulos que constituem o sistema;
- Inclusão de funcionalidades associadas com a gestão de frotas como agendamento de fretes, escalonamento de motoristas, geração de alarmes, *etc.*

Bibliografia

- [1] P. D. Groves, *Principles of GNSS, Inertial and Multisensor Integrated Navigation Systems*. Artech House, 2008. [citado na p. v, 7, 8, 9, 10, 13, 18, 19, 20, 22, 23, 32]
- [2] *Interface Specifications - Navstar GPS Space Segment/Navigation User Interfaces*, Revision d ed., GPS Joint Program Office, 2420 Vela Way, Suite 1866, El Segundo, CA 90245-4659 U.S.A., December 2004. [citado na p. v, 13, 69, 76, 77, 79]
- [3] M. G. Soares, “Posicionamento Diferencial em Tempo Útil via Internet,” Ph.D. dissertation, Faculdade de Engenharia da Universidade do Porto (FEUP), 2004. [citado na p. v, 10, 11, 14, 15, 27, 31, 32, 33]
- [4] Design Publishers. SatNews Publishers: Daily Satellite News. [Online]. Available: <http://www.satnews.com/cgi-bin/home.cgi> [citado na p. v, 18]
- [5] Wikimedia Projects and MediaWiki. Wikimedia Commons. [Online]. Available: http://commons.wikimedia.org/wiki/Main_Page [citado na p. v, 19]
- [6] European Space Agency (ESA). (2000) Europe Satellite Navigation Systems. [Online]. Available: <http://www.satellite-navigation.eu> [citado na p. v, 20, 21]
- [7] John Pike. (2000) Global Security: Reliable Space News and Space Information. [Online]. Available: <http://www.globalsecurity.org/space> [citado na p. v, 23]
- [8] Wikimedia Projects and MediaWiki. (2001) Wikipedia, The Free Encyclopedia. [Online]. Available: <http://en.wikipedia.org/wiki> [citado na p. v, 24]
- [9] Federal Aviation Administration. FAA: Home. [Online]. Available: <http://www.faa.gov> [citado na p. v, 25, 26]

- [10] *Holux GM-210 GPS Receiver User's Guide*, Holux Technology Inc., 8F-11, 26, Tai Yuen St., Chu Pei, Hsin Chu, Taiwan, May 2003. [citado na p. v, 35, 36]
- [11] R. S. C. 159, *Minimum Operational Performance Standards for Global Positioning System/Wide Area Augmentation System Airborne Equipment*, revision rtca/do-229c ed., Radio Technical Commission for Aeronautics, 1828 L Street, NW, Suite 805 Washington, DC 20036 U.S.A, November 2001. [citado na p. v, 41, 45, 99]
- [12] R. S. C. 104, *RTCM Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service*, rtc paper 136-2001/sc104-std version 2.3 ed., Radio Technical Commission for Maritime Services, 1800 Diagonal Road, Suite 600 Alexandria, Virginia 22314-2840 U.S.A, August 2001. [citado na p. v, vii, 46, 48, 49, 103, 104, 113]
- [13] T. Davidson, "Processing and Applying Real Time EGNOS Data," Instituto Superior de Engenharia do Porto (ISEP) and Katholieke Hogeschool Sint-Lieven, Tech. Rep., 2007. [citado na p. vi, 59, 98, 99, 100, 101]
- [14] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle, *GNSS - Global Navigation Satellite Systems: GPS, GLONASS, Galileo and more*. Springer-WienNewYork, 2008. [citado na p. 12, 28, 29]
- [15] J. A. da Silva Rocha, "Aquisição e Disponibilização em Tempo Útil dos Dados Brutos de Receptores SiRF," Instituto Superior de Engenharia do Porto (ISEP), Tech. Rep., 2008. [citado na p. 14, 55, 57]
- [16] L. Jacobson, *GNSS Markets and Applications*. Artech House, 2007. [citado na p. 17, 20]
- [17] Japan Aerospace Exploration Agency. JAXA|Quasi-Zenith Satellite-1 MICHIBIKI. [Online]. Available: http://www.jaxa.jp/index_e.html [citado na p. 24]
- [18] European Space Agency (ESA). ESA EGNOS Navigation. [Online]. Available: <http://www.esa.int/esaNA/egnos.html> [citado na p. 28]
- [19] M. G. Soares and M. B. Malheiro, "EGNOS Based Virtual Reference Stations," in *European Conference on the Use of Modern Information and Communication Technologies (ECUMICT)*, 2008. [citado na p. 32]
- [20] H. Landau, U. Vollath, and X. Chen, "Virtual reference station systems," *Journal of Global Positioning Systems*, vol. 1, pp. 137–143, 2002. [Online]. Available: <http://www.cpgps.org/journals.php> [citado na p. 33]

- [21] R. Chen and X. Li, "Virtual differential gps based on sbas signal," *GPS Solutions*, vol. 8, no. 4, pp. 238–244, 2004. [Online]. Available: <http://dx.doi.org/10.1007/s10291-004-0114-6> [citado na p. 33]
- [22] *SiRF Binary Protocol Reference Manual*, Revision 2.2 ed., SiRF Technology Inc., 217 Devcon Drive, San Jose, CA 95112 U.S.A., June 2007. [citado na p. 37, 67, 71]
- [23] J. Stuyvaert, "Accessing, Decoding and Applying EGNOS Data," Instituto Superior de Engenharia do Porto (ISEP) and Katholieke Hogeschool Sint-Lieven, Tech. Rep., 2006. [citado na p. 46, 59, 92, 94]
- [24] H. S. Hopfield, "Two-quartic tropospheric refractivity profile for correcting satellite data," *Journal of Geophysical Research*, vol. 74, no. 18, pp. 4487–4499, 1969. [Online]. Available: <http://dx.doi.org/10.1029/JC074i018p04487> [citado na p. 79]
- [25] W. F. Foy, "Position-location solutions by taylor-series estimation," *IEEE Trans. Aerospace and Electronic Systems*, vol. AES-12, no. 2, March 1976. [citado na p. 79]
- [26] G. I. Jee, J. S. Hong, H. T. Han, and Y. J. Lee, "Enhanced inverted dgps for automatic vehicle location system," *Journal of Astronomy and Space Sciences*, 1998. [citado na p. 84]
- [27] Google. The Google Maps Javascript API V3. [Online]. Available: <http://code.google.com/intl/pt-BR/apis/maps/documentation/javascript/> [citado na p. 120]