
PROJECTO DE SISTEMAS DE NAVEGAÇÃO

Desenvolvimento de um sistema de detecção de intrusão em
áreas restritas baseado em GPS diferencial

Tiago MATIAS **65655**
João MANITO **73096**
Daniel DE SCHIFFART **81479**

Instituto Superior Técnico
Mestrado em Engenharia Aeroespacial
Sistemas de Navegação

Junho de 2018

1 Objectivo

O objetivo do projeto é a implementação de um sistema de posicionamento global derivado do GPS com correções baseadas em estações locais, um sistema comumente apelidado de GPS diferencial, ou DGPS, e a utilização desse mesmo sistema para detetar intrusão em áreas pré-especificadas.

2 Funcionamento

Neste projeto implementámos melhoramentos de GPS através do método conhecido como GPS diferencial. O GPS diferencial é um sistema para corrigir a posição retornada por um recetor GPS convencional, utilizando um recetor de posição fixa pré-determinada para calcular erros nos *pseudoranges* em tempo real.

2.1 GPS Diferencial

A correção é realizada nos valores dos *pseudoranges* calculados pelo recetor local através dos satélites do sistema GPS. Estes erros, que advêm do facto da influência da atmosfera no sinal de GPS não ser constante, são calculados pela diferença dos *pseudoranges* da estação fixa e a distância real desta aos satélites, cuja posição é conhecida pois esta está presente no sinal de GPS. Uma vez calculados estes erros, estes são transmitidos ao recetor local e subtraídos aos *pseudoranges* calculados por este. Calcula-se então a posição utilizando o método de GPS, mas utilizando desta vez os *pseudoranges* corrigidos.

Sendo d_{GS} a distância entre o satélite N e a estação local, e ρ_{GS} o *pseudorange* obtido nesta, temos, para cada satélite, o erro e na equação 1.

$$\begin{bmatrix} e^{(1)} \\ e^{(2)} \\ \vdots \\ e^{(N)} \end{bmatrix} = \begin{bmatrix} d_{GS}^{(1)} - \rho_{GS}^{(1)} \\ d_{GS}^{(2)} - \rho_{GS}^{(2)} \\ \vdots \\ d_{GS}^{(N)} - \rho_{GS}^{(N)} \end{bmatrix} \quad (1)$$

Sendo $\rho_r^{(N)}$ o *pseudorange* obtido pelo receptor local, temos ρ_{*r} como o valor corrigido para o receptor local do *pseudorange*, na equação 2.

$$\begin{bmatrix} \rho_{*r}^{(1)} \\ \rho_{*r}^{(2)} \\ \vdots \\ \rho_{*r}^{(N)} \end{bmatrix} = \begin{bmatrix} \rho_r^{(1)} - e^{(1)} \\ \rho_r^{(2)} - e^{(2)} \\ \vdots \\ \rho_r^{(N)} - e^{(N)} \end{bmatrix} \quad (2)$$

Com estes novos *pseudoranges* podemos calcular a posição do receptor local da mesma forma que seria calculada no caso de GPS normal.

2.2 Verificação de Intrusão de Área

Para determinar a intrusão de um ponto em áreas restritas, foram formuladas equações com base nas coordenadas ECEF calculadas pelo DGPS. Estas equações foram formuladas para três tipos de volumes geométricos distintos: esfera, cilindro e paralelepípedo (caixa).

2.2.1 Esfera

O cálculo da intrusão num volume esférico com base em coordenadas ECEF foi de relativa facilidade, e foi consequentemente a primeira a ser implementada. Fornecendo um ponto (x_c, y_c, z_c) para o centro da esfera e um raio r , para uma coordenada (x_r, y_r, z_r) fornecida pelo recetor, podemos formular que o ponto está dentro da esfera se

$$\|(x_c, y_c, z_c) - (x_r, y_r, z_r)\| \leq r$$

2.2.2 Cilindro

A implementação do cilindro foi feita através de um ponto (x_b, y_b, z_b) correspondente à base, um raio da base r , e uma altura h . A verificação da intrusão foi feita utilizando uma fatia circular do cilindro com a mesma altura em LLH. Utilizando o centro desta mesma fatia (x_f, y_f, z_f) , formula-se que o ponto está dentro do cilindro se

$$\|(x_f, y_f) - (x_r, y_r)\| \leq r$$

$$z_f = z_r$$

2.2.3 Paralelepípedo

A última implementação foi a do paralelepípedo, que foi a menos trivial a realizar. O algoritmo de deteção utilizado requer um paralelepípedo recto. Para tal, trabalhou-se no sistema de coordenadas ENU, que é um sistema cartesiano. Utilizando dois pontos P_1 e P_3 e a altura h do paralelepípedo obtiveram-se oito pontos que definem o paralelepípedo, como se observa na figura 1.

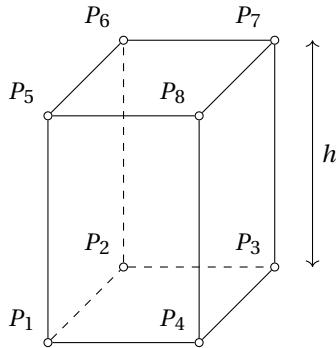


Figura 1: Relação entre os pontos do paralelepípedo.

Utilizando um ponto do receptor $P_r = (x_r, y_r, z_r)$, as condições de intrusão podem ser definidas extraíndo três vetores u , v e w do paralelepípedo.

$$u = P_1 - P_2$$

$$v = P_1 - P_4$$

$$w = P_1 - P_5$$

Com isto definido, o ponto do receptor pode ser considerado dentro do paralelepípedo quando todas as condições das equações 3, 4 e 5 são correspondidas.

$$u \cdot P_1 \leq u \cdot P_r \leq u \cdot P_2 \quad (3)$$

$$v \cdot P_1 \leq v \cdot P_r \leq v \cdot P_4 \quad (4)$$

$$w \cdot P_1 \leq w \cdot P_r \leq w \cdot P_5 \quad (5)$$

2.3 Determinação da posição do receptor

De forma a poder implementar este algoritmo, foi necessário utilizar um receptor de GPS que permita obter os *pseudoranges*, uma vez que o formato NMEA não disponibiliza esses dados. Para o efeito foram usados dois receptores *ublox* EVK-6T.

Usando a informação das especificações IS-GPS-200H e textttu-blox 6 Receiver Description, foi criado um algoritmo que processa as mensagens do receptor (nomeadamente as mensagens RXM-RAW, AID-EPH e AID-HUI). Em seguida foram usados os *pseudoranges*, os dados dos satélites e os parâmetros de correção para obter uma estimativa da posição do receptor usando o método de *least squares*.

3 Implementação e Resultados

A implementação do DGPS foi realizada com recurso ao MATLAB, reutilizando parte do código desenvolvido ao longo do semestre nas aulas da unidade curricular. Numa instância inicial, foram utilizados dados fornecidos pelo professor para as posições dos satélites e para os *pseudoranges* de alguns satélites, obtidos através da estação GNSS colocada no topo do Instituto de Telecomunicações, como descrita no documento *AntennaSurvey.pdf*. Usando esses dados, foi criado um algoritmo de processamento dos mesmos, de forma a converter a informação fornecida pelo receptor de GPS em formatos mais fáceis de utilizar. Posteriormente, esse algoritmo foi usado para determinar os *pseudoranges* dos receptores, que depois foram usados para obter as correções de DGPS. Para a detecção de intrusão, não foi possível utilizar dados de *pseudoranges* reais devido a restrições de tempo. No entanto, validou-se o seu funcionamento utilizando um receptor de GPS com saída NMEA de um *smartphone*.

3.1 Processamento de mensagens de GPS e determinação da posição das estações base

Numa primeira fase, fez-se apenas pós-processamento de dados obtidos pelo receptor em uso. Este receptor apresenta um conjunto de mensagens possíveis na sua especificação, pelo que se optou apenas por processar três tipos de mensagens:

- RXM-Raw, que disponibiliza informação de *pseudoranges* dos vários satélites em uso, bem como um *timestamp* da medição;
- AID-Ephemeris, que disponibiliza informação de efemérides para os vários satélites, bem como informações de integridade dos mesmos;
- AID-HUI, que disponibiliza informação relativa a parâmetros de correção para os relógios e ionosfera.

Recorrendo aos documentos IS-GPS-200H e u-blox 6 Receiver Description, foi criada uma rotina de processamento de dados, que converte o conteúdo de cada mensagem num formato predefinido, igual ao usado no decurso da UC.

De seguida, foram determinadas as posições dos satélites e calculada a posição do receptor usando o método de *Least Squares*. Corrigindo o efeito do *clock bias* e sincronismo dos relógios de GPS, bem como das perturbações troposférica e ionosférica, foi obtida uma estimativa mais precisa para a posição do receptor. Tendo ignorado todas as posições em que o parâmetro PDOP é superior a 4, foi obtida uma estimativa para a posição do receptor. Este processo foi realizado para os dados fornecidos na página da UC (ficheiros ub1.ubx.1744.327600 e ub2.ubx.1744.327600), bem como para um conjunto de dados obtido com o receptor supramencionado.

As posições determinadas são então:

Receptor	Latitude (°)	Longitude (°)	Altitude (m)
UB1	38.7375691156866	9.13852429667141	219.265118060666
UB2	38.7376975066945	-9.13853426280023	209.618087988563

Tabela 1: Posição determinada dos receptores em coordenadas LLH.

Receptor	X (m)	Y (m)	Z (m)
UB1	4918548.05800941	-791216.081612259	3969771.54527795
UB2	4918531.68520011	-791214.325497741	3969776.62639603

Tabela 2: Posição determinada dos receptores em coordenadas ECEF.

Receptor	Erro médio (m)
UB1	26.4846310745313
UB2	30.2741811841672

Tabela 3: Erro médio da posição dos receptores.

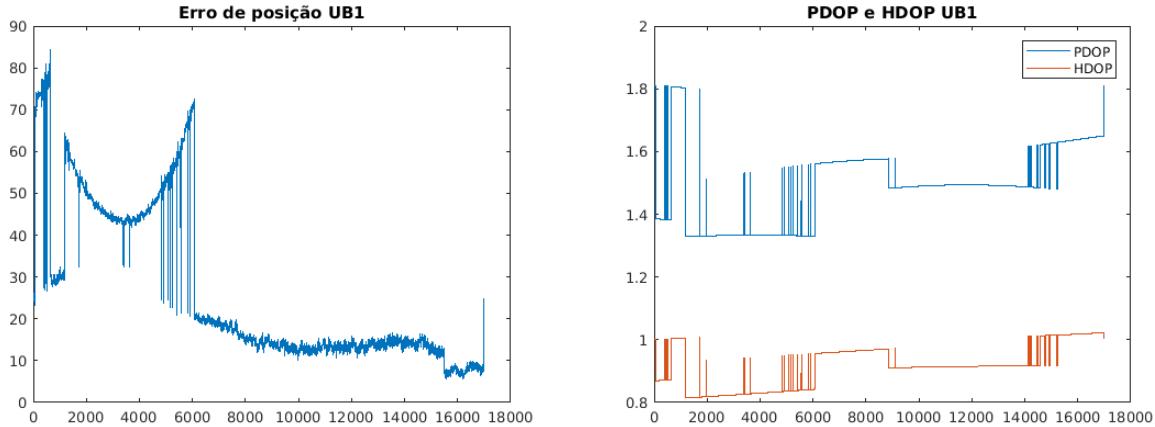


Figura 2: Evolução do erro de posição absoluto, PDOP e HDOP do receptor instalado em UB1 ao longo do período de amostragem.

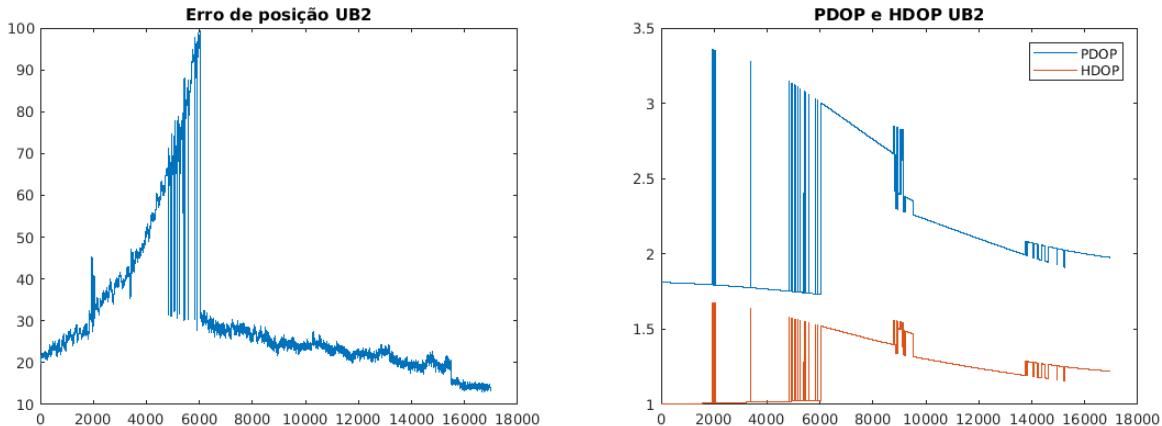


Figura 3: Evolução do erro de posição absoluto, PDOP e HDOP do receptor instalado em UB2 ao longo do período de amostragem.

Dos gráficos, podemos verificar que existe uma zona de elevado erro na posição, comum a ambos os receptores. Embora a causa concreta não tenha sido identificada, presumimos que seja devido a algum erro na nossa implementação. O valor médio foi algo afetado por este fenómeno, tal como se pode verificar no erro médio. Para os gráficos de PDOP e HDOP, verificamos que estes apresentam valores não muito elevados, no caso de UB2, e dentro da gama [0;2.5] no caso de UB1. Há no entanto uma certa correlação entre certos pontos dos gráficos de DOP e erro, em que em tempos semelhantes existem desvios dos valores esperados. Tal dever-se-á possivelmente ao erro de implementação acima referido.

Os resultados desta parte do projecto podem ser obtidos usando o script `determinacao_receptores.m`.

3.2 Determinação das correções diferenciais para os receptores

Como teste do funcionamento do modo diferencial, foram usados os dados dos dois receptores, um como estação base e o outro como receptor móvel. Usando o algoritmo da secção 2.1, e recorrendo a algumas alterações ao mesmo de forma a contabilizar as diferenças de *clock bias* entre receptores e outras correções, obtivemos uma estimativa para a posição do receptor UB2, tendo como estação base o receptor UB1.

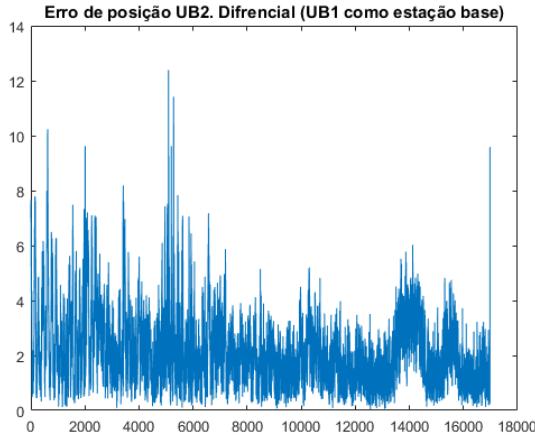


Figura 4: Evolução do erro de posição absoluto de UB2 em modo diferencial ao longo do período de amostragem, tendo como estação base.

Desta figura verificamos que existe um erro claramente inferior ao do método anterior, pelo que podemos afirmar que o método de correção diferencial funciona. É especialmente notável que o comportamento errático presente nas primeiras 6000 amostras analisadas com o método anterior foi completamente anulado, o que reforça a ideia de que a correção diferencial é muito mais robusta que a determinação de posição usando apenas um receptor. No entanto o erro ainda é algo elevado, pelo que há claramente trabalho a desenvolver para permitir reduzir este erro.

Relativamente aos parâmetros PDOP e HDOP, devido a problemas com o código não foi possível calcular os seus valores.

Os resultados desta parte do projecto podem ser obtidos usando o *script* `determinacao_diferencial.m`.

3.3 Detecção de intrusão

A última parte deste projecto consiste na aplicação dos algoritmos e código desenvolvidos nas secções anteriores e aplicá-los a um algoritmo que determine se o receptor se encontra dentro da área restrita ou não. Infelizmente, por restrições de tempo não nos foi possível utilizar a determinação de posição usando DGPS para este efeito. Assim, recorremos ao uso de dados NMEA para essa detecção, de forma a validar o algoritmo de detecção.

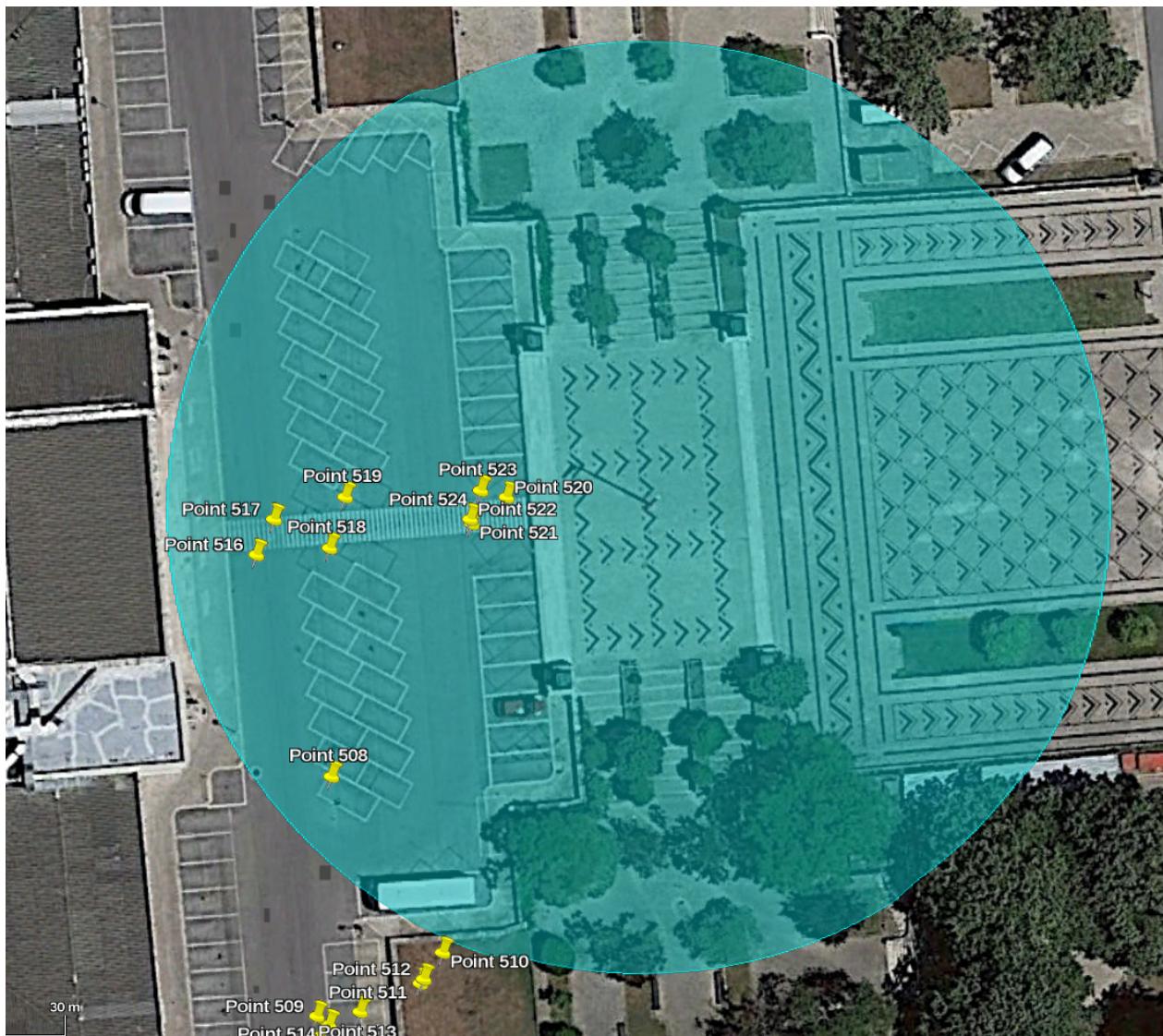


Figura 5: Zona de intrusão esférica.

Na região esférica (aqui representada como uma projecção em 2D) verificamos que o algoritmo detecta corretamente que o receptor de GPS se encontra dentro da zona restrita. Note-se também a falta do ponto 519, que, embora aparente estar dentro da região nesta projecção 2D, se encontra a uma altitude suficientemente elevada para estar fora da zona restrita. Os pontos que se encontram na zona restrita são então $\{517, 518, 520, 521, 522, 523, 524\}$.

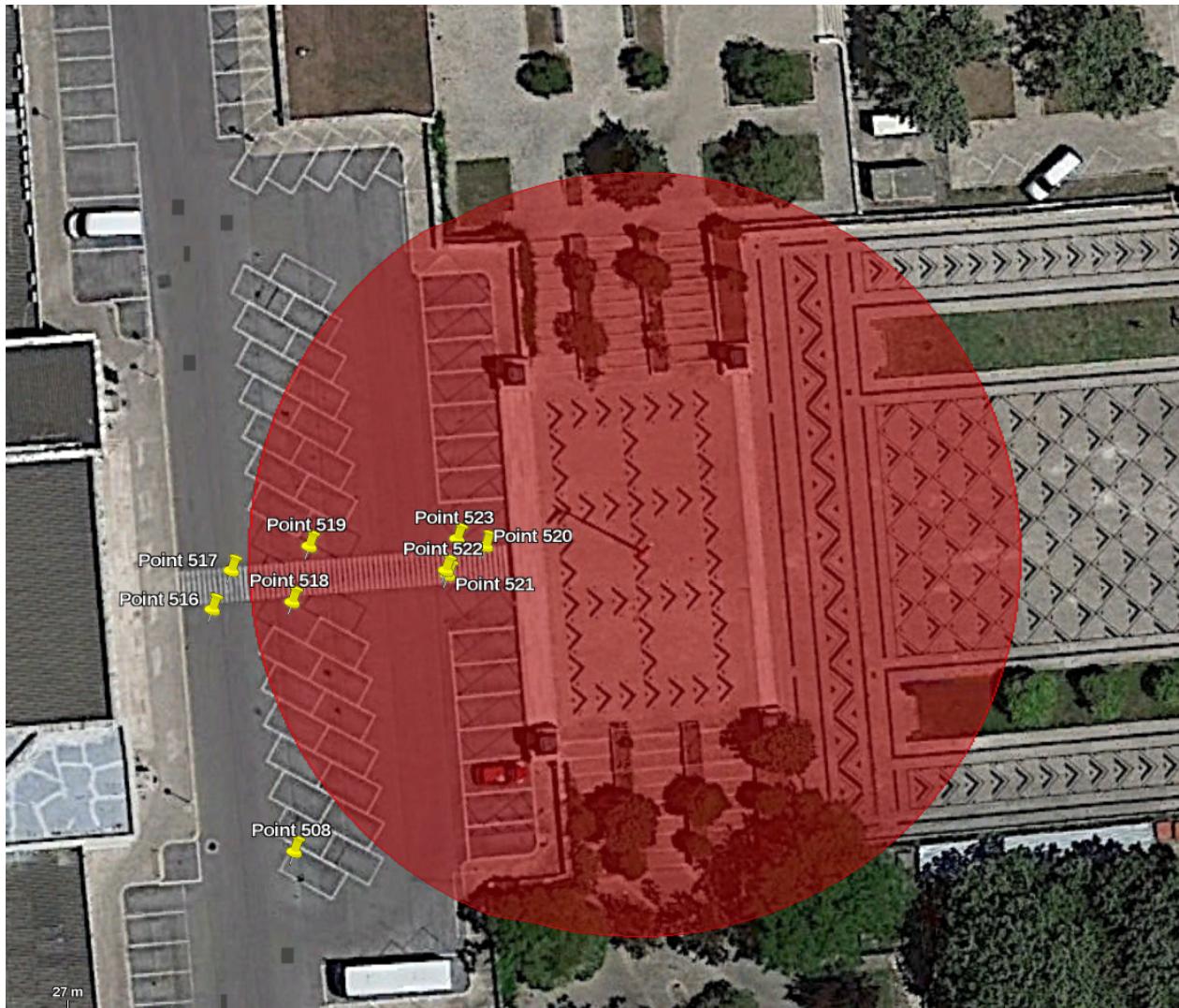


Figura 6: Zona de intrusão cilíndrica.

Na região cilíndrica (aqui representada como uma projecção em 2D) verificamos que o algoritmo detecta corretamente que o receptor de GPS se encontra dentro da zona restrita. Neste caso, devido à forma do cilindro, o ponto 519 já faz parte do conjunto de pontos na zona restrita, sendo estes $\{518, 519, 520, 521, 522, 523, 524\}$.



Figura 7: Zona de intrusão em paralelepípedo.

Na região paralelepípédica, verificamos também que todos os pontos dentro da mesma são corretamente identificados, sendo o conjunto de pontos $\{492, 493, 494, 495, 496, 497, 498, 499, 500, 501\}$.

Os resultados desta parte do projecto podem ser obtidos usando o script `intrusion_test.m`.

4 Conclusões

Embora o projecto não tenha sido concluído (ficando a faltar a utilização do algoritmo de DGPS aplicado a receptores que não os instalados na torre bem como o funcionamento dos algoritmos em tempo real), foi-nos possível verificar que o algoritmo desenvolvido para DGPS apresenta uma redução do erro significativa relativamente à determinação de posição utilizando apenas um receptor de GPS. O erro obtido, embora ainda seja na ordem dos metros, já é pequeno o suficiente para considerar que este algoritmo, mesmo aplicado a um receptor de GPS com taxa de actualização de 1s, é viável para o objectivo principal do projecto, a detecção de intrusão, visto esta ser uma área em que o conhecimento da posição com precisão abaixo da ordem do metro é desnecessário.

Já relativamente ao algoritmo de detecção de intrusão, verificamos que, nos casos testados, este funciona de forma satisfatória, detectando correctamente os pontos em que há intrusão.

Embora não nos tenha sido possível a conclusão do projecto, consideramos que os resultados obtidos são satisfatórios, uma vez que mostram a viabilidade dos algoritmos desenvolvidos para o uso em questão.