

Sistema de Navegación Astronómica por Triangulación (ANST)

17 de agosto de 2025

Resumen

Este informe presenta el Sistema de Navegación Astronómica por Triangulación (ANST), una metodología geométrica y algorítmica para estimar la posición relativa de un observador a partir de mediciones angulares respecto a tres cuerpos celestes de referencia. ANST formaliza una identidad angular fundamental y la incorpora como restricción en esquemas de estimación basados en *bearing-only* y en modos asistidos por efemérides. Se desarrolla la formulación matemática, se proponen algoritmos (modo efemérides-asistido y modo autónomo), se incluye implementación en Python para pruebas numéricas y se analiza la observabilidad y sensibilidad ante ruido y configuraciones degeneradas.

Índice

1. Introducción	3
2. Antecedentes y motivación	3
3. Metodología y objetivos	3
4. Desarrollo	3
4.1. Definición del problema	3
4.2. Construcción geométrica (ANST)	4
4.3. Identidad fundamental	4
4.4. Interpretación práctica	4

5. Formalización matemática	5
5.1. Notación y observaciones	5
5.2. Residuales y función de coste (modo asistido)	5
5.3. Jacobiano (esbozo)	5
6. Algoritmos de estimación	6
6.1. Modo A: Efemérides-asistido (recomendado)	6
6.2. Modo B: Autónomo ANST	6
7. Implementación y validación numérica	6
8. Análisis de observabilidad y sensibilidad	8
8.1. Configuraciones degeneradas	8
8.2. Estimación de covarianzas	9
9. Recomendaciones prácticas	9
10. Conclusiones y trabajo futuro	9

1. Introducción

La navegación en el espacio profundo y la localización relativa de vehículos espaciales requieren técnicas robustas que funcionen incluso con información parcial. ANST nace como una estrategia angular —puramente geométrica— que explota relaciones entre perpendiculares y direcciones aparentes entre tres cuerpos de referencia y un observador. El propósito de este informe es definir ANST formalmente, presentar algoritmos prácticos para su uso en navegación y validar su comportamiento mediante un esquema de simulación numérica.

2. Antecedentes y motivación

La navegación oceánica y astronómica ha usado tradicionalmente medidas angulares (estrellas, sol, planetas) y técnicas de triangulación. En el contexto moderno, los sensores ópticos/estelares y las cámaras permiten obtener *bearings* de alta precisión. ANST propone una identidad angular simple que actúa como término de cierre geométrico, complementando mediciones de bearing y efemérides para mejorar estimaciones y detectar inconsistencias.

3. Metodología y objetivos

Objetivos concretos:

- Formalizar la identidad geométrica central de ANST y su interpretación en el plano y en la esfera celeste.
- Integrar ANST en un esquema de estimación para determinar la posición de un observador D .
- Proveer una implementación reproducible en Python para pruebas y Monte Carlo.
- Analizar condiciones de degeneración, sensibilidad al ruido y recomendaciones operativas.

4. Desarrollo

4.1. Definición del problema

Consideramos cuatro cuerpos:

A (referencia), B , C (masa estelar típica), D (observador).

El objetivo es estimar \mathbf{r}_D (posición de D en un marco inercial local) a partir de mediciones de direcciones aparentes (vectores unitarios) hacia A, B, C y, si están disponibles, efemérides aproximadas de A, B, C .

4.2. Construcción geométrica (ANST)

En el plano definido por A, B, C definimos:

$$\mathbf{u}_{AB} = \frac{\mathbf{r}_B - \mathbf{r}_A}{\|\mathbf{r}_B - \mathbf{r}_A\|}, \quad \mathbf{u}_{AC} = \frac{\mathbf{r}_C - \mathbf{r}_A}{\|\mathbf{r}_C - \mathbf{r}_A\|}, \quad \mathbf{u}_{BC} = \frac{\mathbf{r}_C - \mathbf{r}_B}{\|\mathbf{r}_C - \mathbf{r}_B\|}.$$

Sea \mathbf{n}_1 una unidad perpendicular a \mathbf{u}_{AB} dentro del plano ABC , y análogamente \mathbf{n}_2 perpendicular a \mathbf{u}_{AC} en el mismo plano. Definimos:

$$e_1 = \angle(\mathbf{u}_{BC}, \mathbf{n}_1), \quad e_2 = \angle(\mathbf{u}_{BC}, \mathbf{n}_2), \quad \theta = \angle(\mathbf{n}_1, \mathbf{n}_2) = \angle BAC.$$

4.3. Identidad fundamental

Por construcción geométrica (rotaciones de 90° en el mismo sentido) se cumple:

$$e_2 = |e_1 - \theta|$$

equivalente a la forma trigonométrica:

$$\cos e_2 = \cos(e_1 - \theta) = \cos e_1 \cos \theta + \sin e_1 \sin \theta.$$

En forma vectorial, usando productos escalares:

$$\cos e_1 = \mathbf{u}_{BC} \cdot \mathbf{n}_1, \quad \cos e_2 = \mathbf{u}_{BC} \cdot \mathbf{n}_2, \quad \cos \theta = \mathbf{n}_1 \cdot \mathbf{n}_2.$$

4.4. Interpretación práctica

- **Modo asistido por efemérides:** si $\mathbf{r}_A, \mathbf{r}_B, \mathbf{r}_C$ son conocidos (con incertidumbre limitada), computamos $\mathbf{u}_{BC}, \mathbf{n}_1, \mathbf{n}_2, \theta$ en inercial y usamos ANST como término de regularización en la estimación de \mathbf{r}_D .
- **Modo autónomo ANST:** si solo se tienen bearings desde D en distintos tiempos, ANST añade restricciones geométricas que, con redundancia temporal y suposiciones de escala/planaridad, permiten recuperar posiciones relativas hasta una similitud geométrica (escala/rotación).

5. Formalización matemática

5.1. Notación y observaciones

Denotamos por \mathbf{r}_X la posición inercial de $X \in \{A, B, C, D\}$. Desde D se miden bearings (vectores unitarios):

$$\hat{\mathbf{a}}(t) = \frac{\mathbf{r}_A - \mathbf{r}_D(t)}{\|\mathbf{r}_A - \mathbf{r}_D(t)\|}, \quad \hat{\mathbf{b}}(t) = \frac{\mathbf{r}_B - \mathbf{r}_D(t)}{\|\mathbf{r}_B - \mathbf{r}_D(t)\|}, \quad \hat{\mathbf{c}}(t) = \frac{\mathbf{r}_C - \mathbf{r}_D(t)}{\|\mathbf{r}_C - \mathbf{r}_D(t)\|}.$$

5.2. Residuales y función de coste (modo asistido)

Para una candidata $\tilde{\mathbf{r}}_D$ definimos residuales de bearing y residual ANST. Sea $\hat{\mathbf{a}}^{\text{pred}}(\tilde{\mathbf{r}}_D)$ la predicción del bearing hacia A y análogos. Definimos

$$\mathbf{r}_{\hat{a}} = \hat{\mathbf{a}}^{\text{obs}} - \hat{\mathbf{a}}^{\text{pred}}, \quad \text{etc.}$$

y el residual ANST de cierre

$$r_{\text{ANST}} = \cos e_2^{\text{obs}} - \cos(e_1^{\text{pred}} - \theta),$$

donde $e_1^{\text{pred}}, e_2^{\text{pred}}$ se obtienen con las direcciones iniciales calculadas desde $\mathbf{r}_A, \mathbf{r}_B, \mathbf{r}_C$ y, si se desea, las predicciones desde $\tilde{\mathbf{r}}_D$.

La función de coste a minimizar (por tiempo/instantes) es:

$$J(\tilde{\mathbf{r}}_D) = \sum_t \left(\|\mathbf{r}_{\hat{a}}(t)\|^2 + \|\mathbf{r}_{\hat{b}}(t)\|^2 + \|\mathbf{r}_{\hat{c}}(t)\|^2 + \lambda r_{\text{ANST}}(t)^2 \right),$$

con $\lambda \geq 0$ regulando la confianza en ANST.

5.3. Jacobiano (esbozo)

El jacobiano de la predicción de bearing respecto a $\tilde{\mathbf{r}}_D$ es

$$\frac{\partial}{\partial \tilde{\mathbf{r}}_D} \left(\frac{\mathbf{r}_A - \tilde{\mathbf{r}}_D}{\|\mathbf{r}_A - \tilde{\mathbf{r}}_D\|} \right) = -\frac{1}{\rho} (I - \hat{\mathbf{a}}^{\text{pred}} \hat{\mathbf{a}}^{\text{pred}T}), \quad \rho = \|\mathbf{r}_A - \tilde{\mathbf{r}}_D\|,$$

y el jacobiano del término ANST puede calcularse por diferenciación en cadena o por diferenciación numérica si se prioriza la simplicidad de implementación.

6. Algoritmos de estimación

6.1. Modo A: Efemérides-asistido (recomendado)

1. Recolectar observaciones de bearings $\hat{\mathbf{a}}, \hat{\mathbf{b}}, \hat{\mathbf{c}}$ en instantes t_i .
2. Obtener $\mathbf{r}_A, \mathbf{r}_B, \mathbf{r}_C$ (efemérides o aproximaciones).
3. Construir J y r para la función de coste $J(\tilde{\mathbf{r}}_D)$.
4. Minimizar por Levenberg–Marquardt / `scipy.optimize.least_squares`.
5. Validar residuo ANST y rechazar configuraciones degeneradas.

6.2. Modo B: Autónomo ANST

Problema no lineal simultáneo:

$$\min_{\mathbf{r}_A, \mathbf{r}_B, \mathbf{r}_C, \{\mathbf{r}_D(t)\}} \sum_t \left\| \hat{\mathbf{a}}^{\text{obs}}(t) - \hat{\mathbf{a}}^{\text{pred}}(t) \right\|^2 + \lambda \sum_t r_{\text{ANST}}(t)^2,$$

sujeto a anclajes (p.ej. fijar $\mathbf{r}_A = (0, 0, 0)$) para resolver la ambigüedad de escala. Requiere redundancia temporal y buenas inicializaciones.

7. Implementación y validación numérica

A continuación se incluye una implementación en Python (NumPy + SciPy) para el modo efemérides-asistido. El código es funcional —puedes copiarlo y ejecutarlo en tu entorno. Implementa el cálculo de r_{ANST} según la formulación vectorial descrita.

Código Python (ejecutable, usar numpy y scipy):

```
# ANST_solver.py
import numpy as np
from scipy.optimize import least_squares

def unit(v):
    return v / np.linalg.norm(v)

def compute_plane_basis(uAB, uAC):
    # normal to plane
```

```

n_plane = np.cross(uAB, uAC)
n_plane = n_plane / np.linalg.norm(n_plane)
# n1 = rotate uAB by +90deg inside plane = cross(n_plane, uAB)
n1 = np.cross(n_plane, uAB)
n1 = n1 / np.linalg.norm(n1)
# n2 = rotate uAC by +90deg inside plane
n2 = np.cross(n_plane, uAC)
n2 = n2 / np.linalg.norm(n2)
return n_plane, n1, n2

def anst_residual_inertial(rA, rB, rC):
    uAB = unit(rB - rA)
    uAC = unit(rC - rA)
    uBC = unit(rC - rB)
    _, n1, n2 = compute_plane_basis(uAB, uAC)
    # compute angles via dot products
    cos_e1 = np.clip(np.dot(uBC, n1), -1.0, 1.0)
    cos_e2 = np.clip(np.dot(uBC, n2), -1.0, 1.0)
    e1 = np.arccos(cos_e1)
    e2 = np.arccos(cos_e2)
    theta = np.arccos(np.clip(np.dot(n1, n2), -1.0, 1.0))
    r_anst = np.cos(e2) - np.cos(e1 - theta)
    return r_anst

def bearing_residuals(rD_vec, rA, rB, rC, obs_a, obs_b, obs_c, lam):
    rD = rD_vec.reshape(3)
    a_pred = unit(rA - rD)
    b_pred = unit(rB - rD)
    c_pred = unit(rC - rD)
    # stacking bearing residuals (3-vectors each)
    res = np.hstack([obs_a - a_pred, obs_b - b_pred, obs_c - c_pred])
    # ANST residual (inertial calculation)
    r_anst = anst_residual_inertial(rA, rB, rC)
    # add scaled ANST residual
    res = np.hstack([res, lam * np.array([r_anst])])
    return res

```

```

# Example usage:

if __name__ == "__main__":
    np.random.seed(0)
    # True inertial positions (example)
    rA = np.array([0.0, 0.0, 0.0])
    rB = np.array([1.2, 0.2, 0.0])
    rC = np.array([0.3, 1.0, 0.0])
    rD_true = np.array([2.1, -0.4, 0.7])

    # generate noisy bearings (single epoch)
    obs_a = unit(rA - rD_true) + 1e-3*np.random.randn(3)
    obs_b = unit(rB - rD_true) + 1e-3*np.random.randn(3)
    obs_c = unit(rC - rD_true) + 1e-3*np.random.randn(3)
    obs_a = unit(obs_a); obs_b = unit(obs_b); obs_c = unit(obs_c)

    # inversion
    r0 = np.array([1.5, 0.0, 0.0]) # initial guess
    lam = 1.0 # weight ANST term
    res = least_squares(bearing_residuals, r0, args=(rA,rB,rC,obs_a,obs_b,obs_c, lam))
    print("True D:", rD_true, "Estimated D:", res.x)

```

Notas sobre la implementación

- La función `compute_plane_basis` construye $\mathbf{n}_1, \mathbf{n}_2$ mediante producto vectorial, garantizando que son perpendiculares a $\mathbf{u}_{AB}, \mathbf{u}_{AC}$ dentro del plano ABC .
- El término ANST usado aquí es inercial (calcula r_{ANST} desde posiciones conocidas de A, B, C); si se desea, puede construirse una versión *vista desde D* usando predicciones de bearings.
- Para robustez real, añadir ponderaciones adaptativas, modelos de ruido y verificaciones de condición numérica.

8. Análisis de observabilidad y sensibilidad

8.1. Configuraciones degeneradas

- $\theta \approx 0$ o π (colinealidad de A, B, C): mala condicionamiento, ANST no informativo.

- Cuando B y C aparecen muy próximos desde D : jacobiano de bearings mal condicionado.
- Modo autónomo: ambigüedad de escala (similitud geométrica) sin medidas de distancia absolutas.

8.2. Estimación de covarianzas

Aproximando la minimización por un método Gauss–Newton, la covarianza del estimador se puede estimar como

$$\text{Cov}(\hat{\mathbf{r}}_D) \approx \sigma^2 (J^\top W J)^{-1},$$

con J jacobiano del vector de residuales y σ^2 varianza del ruido en bearings. Esto permite cuantificar precisión esperada y optimizar geometrías observacionales (maximizar la condición de J).

9. Recomendaciones prácticas

- Evitar configuraciones con θ cercano a 0 o π ; elegir tripletas de cuerpos con buena separación angular.
- Recolectar observaciones en múltiples instantes para romper ambigüedades y mejorar precisión.
- Integrar IMU/odometría si está disponible para mejorar seguimiento temporal.
- Implementar robustez frente a outliers (RANSAC, Huber).
- Ajustar el parámetro de regularización λ por validación cruzada (balance entre mediciones y la restricción ANST).

10. Conclusiones y trabajo futuro

ANST presenta una identidad geométrica simple y poderosa que, integrada como restricción en problemas de *bearing-only* o en esquemas asistidos por efemérides, mejora la coherencia geométrica y puede reducir el sesgo en la estimación de posiciones relativas. Los siguientes pasos recomendados:

1. Implementación completa con jacobianos analíticos y pruebas de Monte Carlo para elegir λ óptimo.

2. Extender el esquema a más de tres referencias y a navegación cooperativa entre plataformas.
3. Validación en hardware-in-the-loop con cámaras estelares simuladas y ruido realista.
4. Publicación técnica y apertura de código para reproducibilidad.

Anexo: Diagrama ilustrativo

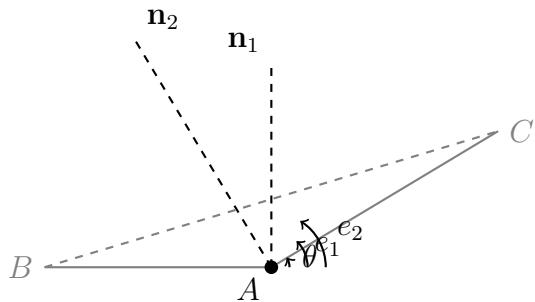


Figura 1: Esquema del sistema ANST en el plano ABC . Las marcas angulares son esquemáticas para ilustración.

Referencias (seleccionadas)

- Fundamentals of Astrodynamics and Applications — D. Vallado (lecturas recomendadas sobre navegación celestial).
- Papers sobre *bearing-only* localization y estimación no lineal (consultar bibliografía específica según interés).
- Autores y agradecimientos: Profesorado y alumnado del Liceo 1, Las Piedras, Uruguay: Ezequiel Souto y Ezequiel Garibaldi.

Documento preparado para revisión interna de LASU.