TREBALL DE FI DE GRAU

Grau en Matemàtiques

Títol: Estimació i navegació robòtica amb IMU en entorns interiors bidimensionals

Autora: Idril-Tadzio Geer Cousté

Directors: Joan Vallvé i Joan Solà

Departament: Institut de Robòtica i Informàtica Industrial (CSIC-UPC)

Convocatòria: 2020-2021

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

Universitat Politècnica de Catalunya

grau en matemàtiques

Treball de fi de grau

Estimació i navegació robòtica amb IMU en entorns interiors bidimensionals

Idril-Tadzio Geer Cousté

Directors:
Joan Vallvé i Joan Solà

3 de maig de 2021

Estimació i navegació robòtica amb IMU en entorns interiors bidimensionals

per Idril-Tadzio Geer Cousté

Grau:
Matemàtiques

Aquest treball ha estat desenvolupat a:
Institut de Robòtica i Informàtica Industrial (CSIC-UPC).

Directors:
Joan Vallvé i Joan Solà

La versió més recent d'aquest treball i material audiovisual adicional poden ser trobats a:
www.idrilgeer.com.

Resum

Estimació i navegació robòtica amb IMU en entorns interiors bidimensionals

per Idril-Tadzio Geer Cousté

Una cosa tant natural per nosaltres com percebre el nostre entorn i localitzar-nos-hi, és en realitat una tasca d'enorme complexitat per una màquina. En la nostra percepció hi entren tots els nostres sentits, amb diferents graus d'importància; la nostra visió és el sentit més important per aquesta tasca, però no es pot menysprear la contribució del sentit de l'equilibri, del tacte, de l'oïda , etc. A més a més també hi interactuen sistemes que no reconeixem clàssicament com a "sentits": la propiocepció, la memòria, el reconeixement de formes, etc. Tots aquests sistemes treballen a una per construir dins la nostra ment un mapa de l'entorn, i alhora s'observen els uns als altres permetent-nos detectar si algun no està funcionant correctament. En el cas de que algun dels nostres sentits esdevingui defectuós, som capaços d'ajustar la nostra percepció per continuar funcionant correctament, a vegades fins a límits sorprenents.

En aquest treball descriurem com una màquina pot usar els seus sensors per aproximar aquest procés, simplificant-ho en entorns planars, i centrant-nos en les contribucions d'un sensor de mesurament inercial (IMU). Aquest sensor correspondria al nostre sentit de l'equilibri. Desenvoluparem la teoria de preintegració associada a aquests sensors, per evitar haver de tractar totes les dades a cada canvi en la nostra estimació, evitant així l'alt cost computacional que això comportaria. Utilitzarem teoria de Lie per dotar de formalisme al procediment, i ho implementarem en una llibreria en benefici de la comunitat d'investigadors en robòtica. Utilitzarem aquesta llibreria per tal de dur a terme experiments amb robots reals, i així validar l'ús de IMUs en l'estimació d'estat d'un robot en un entorn bidimensional. Finalment discutirem els resultats dels experiments, i parlarem de futures línies de recerca possibles.

Agraïments

Moltes gràcies a tothom que ha fet possible aquest treball. En primer lloc, gràcies als meus directors Joan Vallvé i Joan Solà, per tot el que m'han ensenyat aquests mesos. Han sigut moltes hores de parlar sobre teoria de Lie i programar (i debugar!) per aplicar la teoria al món real.

Vull agrair també tot el suport que he rebut aquests anys per part de la gent de l'IRI. Se m'ha obert una àrea de treball molt interessant en la qual m'agradaria seguir en el futur. Gràcies Guillem, Júlia i Sergi(s) per les meves primeres experiències en la recerca robòtica. Gràcies Patrick i Ferran per deixar-me trastejar al taller quan volia construir alguna cosa. I gràcies grup del tupper pels dinars entretinguts plens de conversa.

Mai podré agrair prou als meus pares haver-me donat sempre la llibertat per ser qui sóc, per seguir els meus somnis. Per haver estat sempre al meu costat, per molta distància física que ens separi.

Gràcies Anna i Leo per estar al meu costat. A vosaltres us dedico aquest treball.

Índex

Nomenclatura **xi**

1 Introducció **1**
1.1 Motivació . . . 1
1.2 Objectius . . . 1
1.3 Estructura del treball . . . 2

2 SLAM: Simultaneous Localisation and Mapping **3**
2.1 Formulació probabilística . . . 3
2.2 Graph-SLAM . . . 4
2.2.1 SLAM com a graf de factors . . . 6
2.2.2 Optimització iterativa no lineal . . . 7
El mètode de Gauss-Newton i el cas de mínims quadrats . . . 8
L'estructura dispersa del problema SLAM . . . 9
Optimització sobre varietats . . . 9

3 Teoria de Lie **11**
3.1 Introducció . . . 11
3.2 Grups de Lie . . . 11
3.3 Acció d'un grup sobre un conjunt . . . 11
3.4 Espais tangents i l'Àlgebra de Lie . . . 12
3.4.1 Els isomorfismes de l'àlgebra de Lie . . . 13
3.5 L'aplicació exponencial . . . 14
3.6 Els operadors suma i resta . . . 15
3.7 L'aplicació adjunta . . . 17
3.8 Derivades en grups de Lie . . . 18
3.9 Incertesa en varietats, propagació de covariancies . . . 20

4 El sensor IMU i el seu grup de Lie **21**
4.1 IMUs . . . 21
4.2 IMU i Graph-SLAM . . . 22
4.3 Preintregració de les dades. . . . 25
4.3.1 Integració de l'estat en referència absoluta . . . 25
4.3.2 Definicions de les deltes . . . 26
4.3.3 Preintegració incremental de les deltes . . . 28

4.3.4 El cas 2D . . . 29
4.4 El grup de Lie de la IMU . . . 30
4.4.1 Elements del grup de Lie . . . 30
4.4.2 L'espai tangent . . . 31
4.4.3 L'exponencial . . . 31
4.4.4 Preintegració incremental en el grup de Lie . . . 33
4.4.5 Jacobianes . . . 34
4.4.6 Propagació incremental de la covariància . . . 35
4.4.7 La jacobiana de delta respecte el biaix, correcció de les deltes . . . 36
4.5 Pipeline . . . 36

5 WOLF **37**
5.1 Estructura, l'arbre de WOLF . . . 37
5.1.1 Inicialització del problema . . . 39
5.1.2 Processadors . . . 40
Preintegradors de moviment amb autocalibratge . . . 40
5.1.3 Factors . . . 41
5.1.4 Graf i Solver . . . 41
5.2 ROS . . . 42
5.3 El plugin per la IMU . . . 43
5.3.1 Tests Unitaris . . . 44
Moviment rectilini uniformement accelerat . . . 44
Paràbola . . . 45
Moviment Circular . . . 45

6 Experiments **47**
6.1 El robot: Ana-Pioneer 3 . . . 47
6.1.1 Skid-steering . . . 47
6.1.2 LIDAR . . . 48
6.1.3 IMU . . . 49
6.2 Experiment: Petita volta al laboratori . . . 49
6.2.1 Configuració 1: LIDAR + Odometria de rodes + IMU . . . 49
6.2.2 Configuració 2: Odometria de rodes . . . 50
6.2.3 Configuració 3: Odometria de rodes + IMU . . . 53

7 Conclusió **57**
7.1 Experiments pendents . . . 57
7.2 Futures línies d'investigació . . . 58

A Regles de derivació en grups de Lie **59**
A.1 Regla de la cadena . . . 59
A.2 Jacobiana de l'invers . . . 59
A.3 Jacobiana de la composició . . . 60

A.4 Propietats de la jacobiana de la varietat 60
A.5 Jacobiana de Log() 60
A.6 Jacobiana dels operadors suma i resta 61

B Desenvolupaments del capítol 4 **63**
B.1 Preintegració incremental de les dades 63
B.2 L'espai tangent 63
B.3 Expressions tancades de $\mathcal{R}$, $\mathcal{Q}$, $\mathcal{P}$ 64

C Arxius **67**

Nomenclatura

Llista d'abreviacions

SLAM	Simultaneous Localization And Mapping
IRI	Institut de Robòtica i Informàtica industrial
IMU	Inertial Measurement Unit
WOLF	WindOwed Localization Frames
LIDAR	Laser Imaging Detection and Ranging
YAML	YAML Ain't Markup Language

Operadors

$(\cdot)^{1/2}$	Qualsevol factorització matricial tal que: $\mathbf{A} = (\mathbf{A}^{1/2})^\top \mathbf{A}^{1/2}$.
$\|\cdot\|_{\mathbf{\Omega}^{-1}}$	Norma de Mahalanobis amb covariància $\mathbf{\Omega}^{-1}$
$\oplus$, $\ominus$	operadors suma i resta
$\boxplus$	Composició per deltes.
$\boxplus$	Operador per composar estat amb delta.
$\boxminus$	Operador per recuperar la delta entre dos estats.
$(\cdot)^\wedge$	Isomorfisme "hat".
$(\cdot)^\vee$	Isomorfisme "vee".
exp(), log()	Aplicacions exponencial i logaritme per grups de Lie.
Exp(), Log()	Exponencial i logaritme majúscules.
$\mathrm{Ad}_{\mathcal{X}}$	Aplicació adjunta.
$\mathbf{Ad}_{cX}$	Matriu adjunta.
${}^{\mathcal{X}}\frac{D\mathcal{Y}}{D\mathcal{X}}$, $\frac{D\mathcal{Y}}{D\mathcal{X}}$, $\mathbf{J}^{\mathcal{Y}}_{\mathcal{X}}$	Derivada per la dreta en grups de Lie.
${}^{\mathcal{E}}\frac{D\mathcal{Y}}{D\mathcal{X}}$	Derivada per l'esquerra en grups de Lie.
$\bar{(\cdot)}$	Element precorrecció.
$[\cdot]_\times$	Operador antisimètric.

Notació

Convenció general:

a	Escalar.
$\mathbf{a}$	Vector.
$\mathbf{A}$	Matriu.
$a(\cdot)$	Funció.

Notació especifica:

$p(\cdot)$	Funció de probabilitat.
$\mathbf{x}$	Estat.
$\mathbf{z}_k$	Mesura.
$h_k(\mathbf{x})$	Model de mesura.
$\mathbf{e}_k$	Funció d'error.
$\mathbf{\Omega}_k$	Matriu d'informació.
$\mathbf{\Omega}^{-1}$, $\mathbf{\Sigma}$	Matriu de covariància.
$\mathbf{r}_k$	Residu.
$\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$	Elements d'un grup de Lie.
$\mathcal{T}_{\mathcal{X}}\mathcal{M}$	Espai tangent al punt $\mathcal{X}$ de la varietat $\mathcal{M}$.
$\mathfrak{m}$	L'àlgebra de Lie d'un grup de Lie.
$\boldsymbol{\tau}^{\wedge}$	Element de l'àlgebra de Lie.
$\mathbf{a}$	Acceleració.
$\boldsymbol{\omega}$, ω	Velocitat angular.
$\mathbf{p}$	Posició.
$\mathbf{v}$	Velocitat.
$\mathbf{R}$	Orientació en forma de matriu de rotació.
$\mathfrak{a}$	Acceleració local.
$\mathfrak{v}$	Velocitat local.

1 Introducció

En el camp de la robòtica mòbil, SLAM (Simultaneous Localisation And Mapping) és el problema computacional de construir un mapa d'un entorn desconegut mentre alhora s'hi s'estima la posició del robot. En la seva versió més senzilla, es tracta d'estimar la posició del robot i un conjunt de marcadors posicionals de l'àrea (ie. la posició de objectes a l'entorn). A la literatura s'han presentat diverses maneres de representar el problema, com quadrícules de ocupació o reconstruccions denses i semi-denses 3D RGB.

En aquest treball farem servir representacions per grafs, i ens centrarem en els mètodes SLAM sobre grafs de factors, que estan formalment relacionats amb la formulació probabilística del problema. Hi ha dues maneres de tractar el problema: el filtrat i el suavitzat. El filtrat es basa en tractar només la informació propera en el temps, descartant l'antiga o resumint-la en una distribució a priori. El suavitzat, per contra, fa servir tota la informació disponible des de l'inici del problema. Degut als alts requisits de memòria i computació dels mètodes de suavitzat vers als de filtrat, es van desenvolupar molt més aquests segons quan el hardware disponible era més limitat. Avui dia ja som capaços de fer servir de forma eficient mètodes de suavitzat, i seran els mètodes en què ens centrarem.

1.1 Motivació

L'estimació d'estat en robòtica és un camp molt important. Desenvolupar la capacitat d'un robot de saber on és i com és el seu entorn és crucial si volem aconseguir graus alts d'autonomia, o si el robot ha de treballar sota supervisió humana a distància en entorns desconeguts.

La implementació de sensors de mesurament inercial en problemes de SLAM en entorns 3D està ben estudiada a la literatura [Atchuthan, 2018; Fourmy et al., 2019; Forster et al., 2015]. No obstant, el cas més senzill dels entorns 2D pot ser d'interès en robots que es mouen sobre un pla no inclinat, ja que és més senzill, computacionalment més barat, i permet l'ús de sensors més senzills.

1.2 Objectius

El principal objectiu d'aquest treball és implementar a la llibreria WOLF de l'IRI la capacitat de solucionar problemes de SLAM en 2D que utilitzin un sensor IMU. Per desenvolupar aquesta capacitat haurem de:

1. Definir bé el problema SLAM i identificar com es resol.
2. Trobar un formalisme matemàtic que ens permeti descriure el problema en termes que després siguin útils en l'estimació.
3. Desenvolupar una pipeline per fer servir les dades d'una IMU en el problema.
4. Programar una llibreria d'estimació robòtica que permeti la resolució del problema 2D amb IMUs, a partir de la teoria descrita.
5. Validar-ho tot mitjançant experiments empírics, usant un o múltiples robots de l'IRI.

1.3 Estructura del treball

Comencem aquest treball amb dos capítols teòrics que assentaran la base sobre la qual desenvoluparem la teoria de preintegració. El capítol 2 introdueix el problema SLAM i els grafs de factors. El capítol 3 parla de la teoria de Lie i descriu els objectes matemàtics rellevants que provenen d'aquesta branca de les matemàtiques.

La part central del treball està al capítol 4, on es parla dels sensors que fem servir, i hi desenvolupem la teoria de preintegració, que ens permetrà usar-los per resoldre el problema SLAM de manera eficient.

Al capítol 5 parlem del context informàtic en què programem la llibreria, i breument introduïm les classes i arxius que hi hem afegit.

Finalment tenim un capítol dedicat a l'experimentació empírica, on validem la teoria a través del codi aplicat a robots de l'IRI.

2 SLAM: Simultaneous Localisation and Mapping

2.1 Formulació probabilística

Durant la resta d'aquest treball voldrem estimar l'*estat*, donades una sèrie de dades dels sensors del robot. Considerarem l'estat com la col·lecció de tots els aspectes del robot i el seu entorn que ens interessaria estimar ja que podrien tenir efecte en el seu futur. Això inclou la posició i velocitat del robot (i dels seus sensors), els paràmetres intrínsecs dels sensors (com ara paràmetres de calibratge), les posicions dels objectes presents a l'entorn, i altres característiques rellevants al problema concret. Denotarem l'estat per $\mathbf{x} = (\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{l}_1, \mathbf{l}_2, \dots)$, i farem servir la paraula *estat(s)* per referir-nos tant al total $\mathbf{x}$, com a subconjunts o elements del mateix. Per exemple podríem parlar dels estats del robot $\mathbf{x}_i$ o dels estats dels objectes presents al seu entorn $\mathbf{l}_i$. En general no fem aquesta última distinció i notarem per $\mathbf{x}_i$ tots els tipus d'estats.

Els estats poden ser dinàmics com la posició del robot, o estàtics com la posició d'un arbre que el robot ha detectat. En aquest treball l'estat normalment farà referència a combinacions de la pose del robot, la seva velocitat, la posició i característiques d'objectes de l'entorn, etc.

Denotarem per $\mathbf{z}_k$ les mesures que efectuen els sensors, que observen l'estat en un moment donat. Considerarem que cada mesura prové d'un sensor en un moment concret.

L'evolució de l'estat $\mathbf{x}$ està subjecta a lleis probabilístiques i és típicament estocàstica (estats posteriors dependran d'estats anteriors). El problema SLAM, és el d'estimar la probabilitat $p(\mathbf{x}|Z)$, notant com Z el conjunt de totes les mesures preses fins aquest moment. La formula de Bayes ens permet expressar aquesta probabilitat condicional $p(\mathbf{x}|Z)$ com una funció de $p(Z|\mathbf{x})$:

$$p(\mathbf{x}|Z) = \frac{p(Z|\mathbf{x})p(\mathbf{x})}{p(Z)} \tag{2.1}$$

una probabilitat que podrem formular a partir dels models dels nostres sensors. Com que tenim les mesures dels sensors, utilitzarem l'estimació Bayesiana del màxim a posteriori per tal de trobar l'estimador de $\mathbf{x}$ que ens maximitza la probabilitat de l'estat donada la nostra mostra de Z:

Definició 2.1.1: Estimació del màxim a posteriori

Sigui $\boldsymbol{\theta}$ un paràmetre i X un conjunt d'observacions. Donada una distribució a priori $p(\boldsymbol{\theta})$ del paràmetre i una funció de versemblança $p(X|\boldsymbol{\theta})$ per les observacions, podem trobar una distribució a posteriori $p(\boldsymbol{\theta}|X)$ que té en compte les observacions usant la formula de Bayes:

$$p(\boldsymbol{\theta}|X) = \frac{p(X|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(X)} \tag{2.2}$$

L'estimador màxim a posteriori és el paràmetre $\boldsymbol{\theta}^*_{MAP}$ que maximitza $p(\boldsymbol{\theta}|X)$

En el nostre cas, el paràmetre a estimar és l'estat i les observacions són les nostres mesures:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} p(\mathbf{x}|Z) = \arg\max_{\mathbf{x}} \frac{p(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_K|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_K)} \tag{2.3}$$

Una assumpció clau és que les mesures $\mathbf{z}_i$ són independents, cosa que ens permet simplificar i considerar el problema d'optimització:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} p(\mathbf{x}) \prod_{k=1}^{K} p(\mathbf{z}_k|\mathbf{x}) \tag{2.4}$$

Segons creix el problema (entren noves mesures i l'estat canvia) actualitzarem aquesta distribució. En el cas d'un mètode de filtrat, es faria servir com a distribució a priori $p(\mathbf{x})$ la que hagi sortit de l'anterior iteració, i el productori de $p(\mathbf{z}_k|\mathbf{x}_{t_n})$ faria referència a les dades rellevants a l'estat actual. Això permet resumir l'efecte d'estats i dades previs en aquesta distribució a priori i poder descartar-los de la memòria.

Pels mètodes de suavitzat utilitzarem l'estimació de màxima versemblança, que correspon a suposar que la distribució a priori $p(\mathbf{x})$ en l'estimació del màxim a posteriori és uniforme. Donat que en la majoria de problemes de SLAM començarem sense cap informació sobre l'entorn, és una assumpció molt raonable. A més a més, és més recomanable usar el de màxima versemblança si no podem assegurar que la distribució a priori és bona:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \prod_{k=1}^{K} p(\mathbf{z}_k|\mathbf{x}) \tag{2.5}$$

Recordem a més que en un mètode de suavitzat tenim en compte totes les dades i tot l'estat des de l'inici del problema.

2.2 Graph-SLAM

Considerem el cas d'un robot movent-se per un entorn desconegut. Mentre es mou, pren mesures del seu propi moviment i detecta objectes o propietats d'aquest entorn. En la Fig. 2.1 podem veure una situació d'exemple:

1. A temps inicial el rover està en la part inferior esquerra sense saber res del seu entorn.

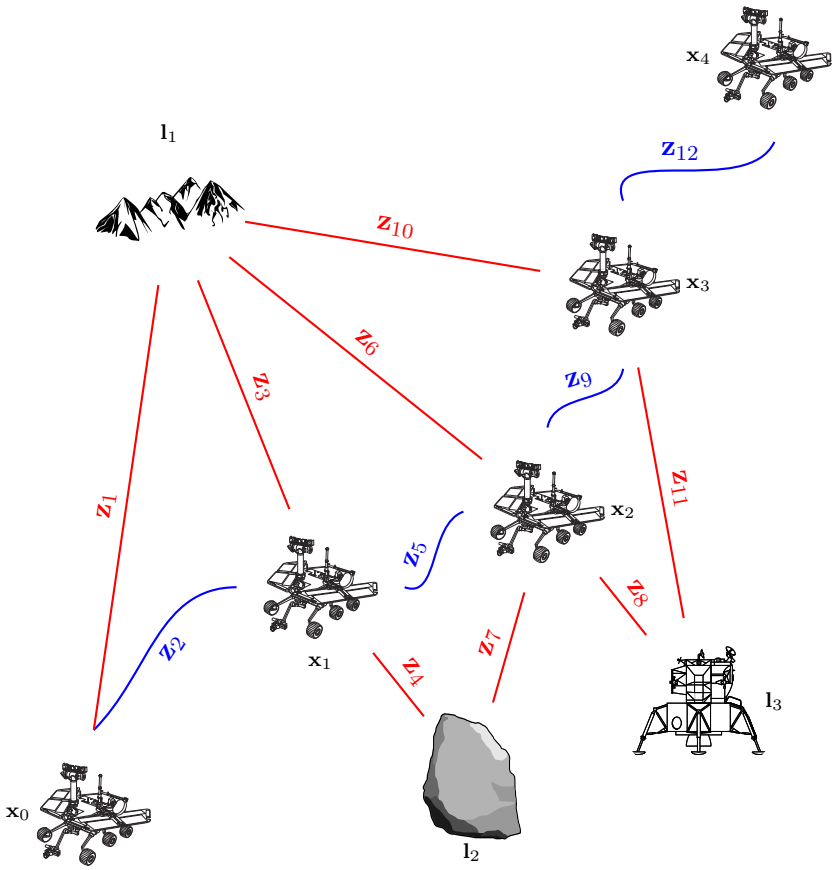

FIGURA 2.1: Rover movent-se per Mart. En blau el moviment, en vermell els sensors detectant objectes de l'entorn. Dibuixos vectorials cortesia de OpenClipArt

L'estat serà simplement $\mathbf{x} = (\mathbf{x}_0)$, corresponent a la posició inicial del rover (que probablement inicialitzarem a 0 a falta de referència global).

2. El rover pren una mesura del seu entorn $\mathbf{z}_1$, que detecta una muntanya a la llunyania. L'estat ara és $\mathbf{x} = (\mathbf{x}_0, \mathbf{l}_1)$, on fem distinció del tipus d'estats per raons il·lustratives.

3. El rover ara es mou durant un temps i torna a prendre una mesura de l'entorn en t_1. Entre t_0 i t_1 ha mesurat el seu moviment[1] en $\mathbf{z}_2$. La mesura de l'entorn detecta dos objectes, que separem en dos mesures $\mathbf{z}_3$ i $\mathbf{z}_4$. Hem afegit a l'estat dos estats: $\mathbf{x}_1$ corresponent a la nova posició del rover i $\mathbf{l}_2$ corresponent a la roca. A més a més, tenim una nova mesura de la muntanya a la llunyania, que ens permet estimar amb més precisió la seva posició real respecte $\mathbf{x}_0$ i $\mathbf{x}_1$. L'estat ara és $\mathbf{x} = (\mathbf{x}_0, \mathbf{x}_1, \mathbf{l}_1, \mathbf{l}_2)$.

4. El rover procedeix d'aquesta manera fins al moment t_4 on ja no detecta res al seu entorn. L'estat final (per ara) és $\mathbf{x} = (\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3)$ amb mesures $Z = (\mathbf{z}_1, \ldots, \mathbf{z}_{12})$

Podem intuir que podrem representar aquesta situació mitjançant un graf que ens relacioni els estats a través de les mesures.

[1] Per exemple, a partir del nombre de revolucions de les rodes.

2.2.1 SLAM com a graf de factors

La probabilitat (2.5) és producte de K factors, tots ells independents. Aquests factors provenen de mesures que observen un nombre reduït d'estats (poses del robot, objectes de l'entorn o combinacions d'aquests). És d'interès representar aquestes relacions en un graf:

Definició 2.2.1: Graf de factors

Un graf de factors és un graf bipartit amb dos tipus de nodes: els variables i els de factor.

- **Node variable:** Representen els estats.
- **Node factor:** Representen les observacions entre els estats.

El graf es construeix unint amb arestes els factors als estats que observen. Els nodes factors codifiquen tota la informació que entra al sistema, mentre les arestes capturaran com es propaga aquesta informació als estats que volem estimar.

La Fig. 2.1 expressada com a graf de factors queda:

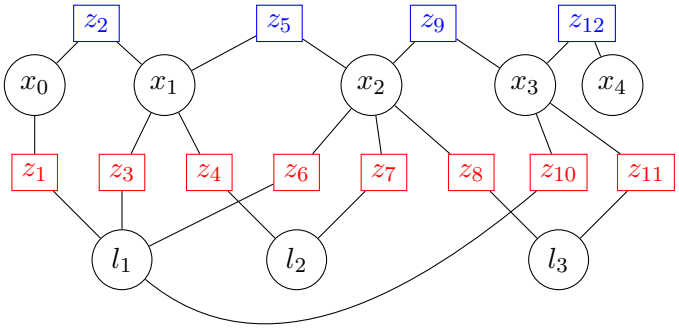

FIGURA 2.2: Els factors blaus corresponen a mesures de moviment i els vermells a mesures de l'entorn. Aquesta distinció és aquí només per qüestions de visualització.

Les probabilitats condicionals que conformen els factors s'obtenen fàcilment dels models de mesura dels sensors:

$$\mathbf{z}_k = h_k(\mathbf{x}) + \mathbf{v}_k, \;\; \mathbf{v}_k \sim \mathcal{N}\{0, \mathbf{\Omega}_k^{-1}\} \tag{2.6}$$

On $\mathbf{v}_k$ és el soroll modelat com una variable aleatòria normal multivariable amb matriu de covariància $\mathbf{\Omega}_k^{-1}$ [2]. Així, els factors del producte de probabilitats queden:

$$p(\mathbf{z}_k|\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{z}_k - h_k(\mathbf{x}))^\top \mathbf{\Omega}_k (\mathbf{z}_k - h_k(\mathbf{x}))\right) \tag{2.7}$$

i definint els errors $\mathbf{e}_k$ com[3]

$$\mathbf{e}_k(\mathbf{x}) = h_k(\mathbf{x}) - \mathbf{z}_k \tag{2.8}$$

[2] Les matrius $\mathbf{\Omega}$ són les matrius d'informació de les mesures, les inverses de les matrius de covariància.
[3] Els errors es poden definir d'altres maneres.

els nostres factors admeten la forma:

$$p(\mathbf{z}_k|\mathbf{x}) = \exp\left(-\frac{1}{2}\mathbf{e}_k^\top \mathbf{\Omega}_k \mathbf{e}_k\right) \tag{2.9}$$

Per cada dada observada $\mathbf{z}_k$ tenim el factor i l'error $\mathbf{e}_k$ associats. La probabilitat condicionada que havíem vist a l'equació (2.5) pot ser ara escrita com el producte de tots els factors:

$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} \exp\left(-\frac{1}{2}\mathbf{e}_k^\top \mathbf{\Omega}_k \mathbf{e}_k\right) \tag{2.10}$$

Maximitzar aquesta funció de distribució de probabilitat és equivalent a minimitzar la seva log-versemblança negativa, també anomenada *cost*:

$$F(\mathbf{x}) := -\ln p(\mathbf{x}|\mathbf{z}) = \frac{1}{2}\sum_{k=1}^{K} \mathbf{e}_k(\mathbf{x})^\top \mathbf{\Omega}_k \mathbf{e}_k(\mathbf{x}) = \sum_{k=1}^{K} F_k \tag{2.11}$$

Observem que els termes F_k són la norma de Mahalanobis $\|\cdot\|_{\mathbf{\Omega}^{-1}}$ al quadrat dels errors $\mathbf{e}_k$ amb matriu de covariància $\mathbf{\Omega}_k^{-1}$:

$$F(\mathbf{x}) = \frac{1}{2}\sum_{k=1}^{K} \|\mathbf{e}_k\|^2_{\mathbf{\Omega}_k^{-1}} \tag{2.12}$$

Definint també el *residu del factor* com

$$\mathbf{r}_k = \mathbf{\Omega}_k^{1/2}\mathbf{e}_k \tag{2.13}$$

on $\mathbf{\Omega}_k^{1/2}$ és qualsevol factorització de la matriu tal que $\mathbf{\Omega}_k = (\mathbf{\Omega}_k^{1/2})^\top \mathbf{\Omega}_k^{1/2}$, la funció de cost admet també la forma equivalent $F(\mathbf{x}) = \frac{1}{2}\sum_{k=1}^{K} \|\mathbf{r}_k\|^2$

2.2.2 Optimització iterativa no lineal

Les nostres funcions d'error seran, en general, no lineals. A més a més, el nostre graf (i.e. el problema) creix a mesura que s'afegeixen nous factors i dades. Per tal de trobar la $\mathbf{x}^*$ que ens minimitza el cost

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} F(\mathbf{x}) \tag{2.14}$$

utilitzarem mètodes d'optimització no lineal iteratius. En aquest treball no entrarem en profunditat en els diferents mètodes que existeixen, tots ells proposen una sèrie de passos $\Delta\mathbf{x}$ tal que $\mathbf{x}_n = \mathbf{x}_{n-1} + \Delta\mathbf{x}_n$ convergeix cap a $\mathbf{x}^*$. En general segueixen el següent esquema:

1. aproximar $F(\mathbf{x})$ al voltant d'una estimació de l'estat $\hat{\mathbf{x}}$ amb una forma analíticament tractable,
2. calcular $\Delta\mathbf{x}$,
3. actualitzar l'estimació usant el pas $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \Delta\mathbf{x}$,

4. iterar fins convergir numèricament.

La nostra funció de cost (2.11) és minimitzable amb un procediment de mínims quadrats.

El mètode de Gauss-Newton i el cas de mínims quadrats

El mètode de Newton aproxima la funció de cost a cada iteració per un paraboloide. S'aproxima la funció de cost per l'expansió de Taylor de segon ordre al voltant de $\hat{\mathbf{x}}$

$$F(\hat{\mathbf{x}} + \Delta\mathbf{x}) \approx F(\hat{\mathbf{x}}) + \nabla F \Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^\top \mathbf{H}_F \Delta\mathbf{x} \tag{2.15}$$

on ∇F és el gradient i $\mathbf{H}_F$ és la hessiana. Diferenciant-la respecte $\Delta\mathbf{x}$ i igualant a zero

$$\nabla F + \mathbf{H}_F \Delta\mathbf{x}^* = 0 \tag{2.16}$$

trobem que el pas òptim $\Delta\mathbf{x}^*$ pel mètode de Newton és

$$\Delta\mathbf{x}^*_N = -\mathbf{H}_F^{-1} \nabla F^\top \tag{2.17}$$

En cas de que la funció de cost sigui de la forma

$$F(\mathbf{x}) = \frac{1}{2}\mathbf{e}(\mathbf{x})^\top \boldsymbol{\Omega} \mathbf{e}(\mathbf{x}) \tag{2.18}$$

amb $\boldsymbol{\Omega}$ una matriu simètrica definida positiva, el vector gradient ∇F i la hessiana $\mathbf{H}_F$ vénen donats per

$$\nabla F = \left(\mathbf{e}^\top \boldsymbol{\Omega} \frac{\partial \mathbf{e}}{\partial \mathbf{x}} \right) \Big|_{\hat{\mathbf{x}}} = \hat{\mathbf{e}}^\top \boldsymbol{\Omega} \mathbf{J} \tag{2.19}$$

$$\mathbf{H}_F = \left(\frac{\partial \mathbf{e}}{\partial \mathbf{x}}^\top \boldsymbol{\Omega} \frac{\partial \mathbf{e}}{\partial \mathbf{x}} + \mathbf{e}^\top \boldsymbol{\Omega} \frac{\partial^2 \mathbf{e}}{\partial \mathbf{x}^2} \right) \Big|_{\hat{\mathbf{x}}} = \mathbf{J}^\top \boldsymbol{\Omega} \mathbf{J} + \hat{\mathbf{e}}^\top \boldsymbol{\Omega} \mathcal{H} \tag{2.20}$$

on $\hat{\mathbf{e}}$, $\mathbf{J}$ i $\mathcal{H}$ són l'error i les seves primeres dues derivades al voltant de l'estimació de l'estat $\hat{\mathbf{x}}$, que ens porta a una aproximació de segon ordre de la forma:

$$F(\hat{\mathbf{x}} + \Delta\mathbf{x}) \approx \frac{1}{2}\hat{\mathbf{e}}^\top \boldsymbol{\Omega} \mathbf{e} + (\hat{\mathbf{e}}^\top \boldsymbol{\Omega} \mathbf{J}) \Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^\top (\mathbf{J}^\top \boldsymbol{\Omega} \mathbf{J} + \hat{\mathbf{e}}^\top \boldsymbol{\Omega} \mathcal{H}) \Delta\mathbf{x} \tag{2.21}$$

El mètode de Gauss-Newton es basa en negligir el terme de segon grau $\hat{\mathbf{e}}^\top \boldsymbol{\Omega} \mathcal{H}$ de la hessiana $\mathbf{H}_F$, cosa que podem fer ja que $\hat{\mathbf{e}}$ és petit. Definint la hessiana aproximada com $\mathbf{H} := \mathbf{J}^\top \boldsymbol{\Omega} \mathbf{J} \approx \mathbf{H}_F$ el pas pel mètode de Gauss-Newton queda:

$$\Delta\mathbf{x}^*_{GN} = -\mathbf{H}^{-1} \nabla F^\top = -\mathbf{J}^+_{\boldsymbol{\Omega}} \hat{\mathbf{e}} \tag{2.22}$$

on el terme $\mathbf{J}^+_{\boldsymbol{\Omega}} := (\mathbf{J}^\top \boldsymbol{\Omega} \mathbf{J})^{-1} \mathbf{J}^\top \boldsymbol{\Omega}$ es coneix com la *inversa generalitzada per l'esquerra amb pes* de la jacobiana $\mathbf{J}$, i veiem que el pas és proporcional a l'error observat.

L'estructura dispersa del problema SLAM

En el nostre problema, la funció de cost és una suma de distàncies de Mahalanobis al quadrat, i per tant és equivalent a (2.18). Cada error $\mathbf{e}_k$ ve de una mesura $\mathbf{z}_k$ amb un soroll associat de covariància $\mathbf{\Omega}_k^{-1}$ i correspon a un factor del graf. La minimització de la nostra funció de cost fent servir Gauss-Newton es fa linealitzant els errors $\mathbf{e}_k$, resolent el problema de mínims quadrats, i iterant fins a convergir:

$$\mathbf{e}_k(\hat{\mathbf{x}} + \Delta\mathbf{x}) \approx \hat{\mathbf{e}}_k + \mathbf{J}_k\Delta\mathbf{x} \tag{2.23}$$

on $\hat{\mathbf{e}}_k := \mathbf{e}_k(\hat{\mathbf{x}})$ i $\mathbf{J}_k = \frac{\partial \mathbf{e}_k(\mathbf{x})}{\partial \mathbf{x}}\Big|_{\hat{\mathbf{x}}}$ [4], i les matrius Jacobiana i Hessiana del mètode de Gauss-Newton queden:

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_1 \\ \vdots \\ \mathbf{J}_K \end{bmatrix} \tag{2.24}$$

$$\mathbf{\Omega} = \text{diag}(\mathbf{\Omega}_1, \dots, \mathbf{\Omega}_K) \tag{2.25}$$

$$\mathbf{H} = \mathbf{J}^\top\mathbf{\Omega}\mathbf{J} = \sum_{k=1}^{K} \mathbf{J}_k^\top\mathbf{\Omega}_k\mathbf{J}_k \tag{2.26}$$

En els problemes de SLAM aquestes matrius són disperses, ja que les $\mathbf{J}_k$ ho són per construcció ($h_k(\mathbf{x})$ involucra només una petita part de $\mathbf{x}$). Aquesta dispersió fa que puguem aplicar mètodes de factorització (per exemple Cholesky o factorització QR) molt eficients per solucionar (2.22).

Optimització sobre varietats

Ens interessa poder optimitzar l'estat en espais on tot ell (o part) no està representat en un espai euclidià.

Definició 2.2.2: Espai Euclidià

Un espai euclidià és un espai afí sobre els reals tal que el seu espai vectorial associat és de dimensió finita i dotat d'un producte interior.
Una propietat important és que tot espai euclidià de dimensió n és isomorf a $\mathbb{R}^n$

Per exemple les orientacions en un espai 3D no són euclidianes, ja que $SO(3) \simeq \mathbb{P}^3(\mathbb{R}) \ncong \mathbb{R}^3$. De la mateixa manera $SO(2) \simeq \mathbb{S}^1 \ncong \mathbb{R}^2$.

En casos així, un procediment comú és el de fer l'optimització sobre la varietat definida per les restriccions sobre l'espai. Un exemple seria S^1 que es pot definir amb una restricció sobre $\mathbb{C}$: $S^1 = \{z \in \mathbb{C} \mid |z| = 1\}$.

[4] en el cas que definim l'error com a (2.8): $\mathbf{J}_k = \frac{\partial h_k(\mathbf{x})}{\partial \mathbf{x}}\Big|_{\hat{\mathbf{x}}}$. D'altra manera cal calcular la jacobiana amb la nova expressió de l'error

La idea és representar el pas de correcció sobre un espai euclidià (el tangent) mentre mantenim l'estat en l'espai original (que defineix la varietat).

Necessitarem definicions mínimes de les $\Delta\mathbf{x}$ de manera que el mínim de la funció de cost (2.11) s'assoleixi en un únic punt. Definirem doncs les $\Delta\mathbf{x}$ en un espai euclidià tangent a la varietat en el punt definit per l'estat actual. En aquesta situació no ens convé la composició additiva que fèiem servir, $\mathbf{x} = \hat{\mathbf{x}} + \Delta\mathbf{x}$ ja que pressuposa espais euclidians de la mateixa dimensió i podria no estar ben definida, o el resultat de la composició podria no estar sobre la varietat. Definirem, doncs, un nou operador $\oplus$ que ens porti variacions en l'espai tangent euclidià a variacions locals sobre la varietat, $\Delta\mathbf{x} \longrightarrow \hat{\mathbf{x}} \oplus \Delta\mathbf{x}$ de manera que $\mathbf{x} = \hat{\mathbf{x}} \oplus \Delta\mathbf{x}$ estigui ben definida i caigui sobre la varietat.

Aquesta nova definició afecta a la linealització dels errors (2.23), que esdevé $\mathbf{e}_k(\mathbf{x}) = \mathbf{e}_k(\hat{\mathbf{x}} \oplus \Delta\mathbf{x}) \approx \hat{\mathbf{e}}_k + \mathbf{J}'_k\Delta\mathbf{x}$ amb $\hat{\mathbf{e}}_k$ com abans i $\mathbf{J}'_k = \left.\frac{\partial \mathbf{e}_k(\mathbf{x})}{\partial \Delta\mathbf{x}}\right|_{\hat{\mathbf{x}}}$ una jacobiana que haurem de definir bé. La resolució del problema segueix de manera natural tenint en compte la utilització del nou operador de composició al final de cada iteració:

$$\hat{\mathbf{x}} \longleftarrow \hat{\mathbf{x}} \oplus \Delta\mathbf{x}^* \tag{2.27}$$

En el següent capítol introduirem la teoria de Lie que usarem per definir aquestes deltes, l'operador $\oplus$ i les jacobianes.

3 Teoria de Lie

3.1 Introducció

Un grup de Lie és un objecte matemàtic abstracte que sorgeix de la feina del matemàtic Sophus Lie sobre la teoria de grups de transformacions continues. Ens donen una base potent per formular i tractar correctament els problemes d'estimació. Mitjançant la teoria de Lie podem construir amb rigor una base amb la que tractar incertesa, derivades i integrals sobre la representació de l'estat com a varietat.

3.2 Grups de Lie

Un grup de Lie és una varietat diferenciable els elements de la qual satisfan les propietats de grup.

Definició 3.2.1: Varietat diferenciable

La definició formal de varietat és llarga i massa feixuga per al que les farem servir en aquest treball. La idea principal amb la que ens hem de quedar respecte varietats diferenciables és: una varietat diferenciable és un espai topològic que localment és prou similar a un espai lineal.
Ens les podem imaginar com hipersuperfícies corbades i suaus, sense punxes ni plecs, en un espai de dimensió superior.

Definició 3.2.2: Grup

Un grup $(\mathcal{G}, \circ)$ és un conjunt $\mathcal{G}$ amb una operació $\circ$ que $\forall\ \mathcal{X}, \mathcal{Y}, \mathcal{Z} \in \mathcal{G}$ compleix els següents axiomes:

1. **Tancat per la operació:** $\mathcal{X} \circ \mathcal{Y} \in \mathcal{G}$
2. **Element identitat:** $\exists \mathcal{E} \in \mathcal{G}$ tq $\mathcal{E} \circ \mathcal{X} = \mathcal{X} \circ \mathcal{E} = \mathcal{X}$
3. **Element invers:** $\exists \mathcal{X}^{-1}$ tq $\mathcal{X}^{-1} \circ \mathcal{X} = \mathcal{X} \circ \mathcal{X}^{-1} = \mathcal{E}$
4. **Associativitat:** $(\mathcal{X} \circ \mathcal{Y}) \circ \mathcal{Z} = \mathcal{X} \circ (\mathcal{Y} \circ \mathcal{Z})$

La combinació d'aquestes dues definicions ens permetrà fer càlcul i composicions no lineals.

3.3 Acció d'un grup sobre un conjunt

Els grups de Lie poden causar transformacions sobre elements d'un altre conjunt, per exemple rotacions i translacions, propietat que es fa servir molt a robòtica. Aquestes transformacions

vénen donades per l'*acció del grup sobre un conjunt*:

Definició 3.3.1: Accions

Donat un grup $\mathcal{G}$ i un conjunt $\mathcal{V}$ es denota per $\mathcal{X} \cdot v$ l'acció per l'esquerra de $\mathcal{G} \in \mathcal{M}$ sobre $v \in \mathcal{V}$:

$$\cdot : \mathcal{G} \times \mathcal{V} \longrightarrow \mathcal{V} \tag{3.1}$$

$$(\mathcal{X}, v) \mapsto \mathcal{X} \cdot v \tag{3.2}$$

que ha de satisfer les següents propietats:

1. **Identitat:** $\mathcal{E} \cdot v = v$
2. **Compatibilitat:** $(\mathcal{X} \circ \mathcal{Y}) \cdot v = \mathcal{X} \cdot (\mathcal{Y} \cdot v)$

De manera anàloga definim l'acció per la dreta.

Dos exemples comuns de grup actuant sobre vectors són el de les matrius de rotació $SO(n)$ ($\mathbf{R} \cdot \mathbf{x} = \mathbf{R}\mathbf{x}$) i el dels quaternions $\mathcal{S}^3$ ($\mathbf{q} \cdot \mathbf{x} = \mathbf{q}\mathbf{x}\mathbf{q}^*$)

3.4 Espais tangents i l'Àlgebra de Lie

Donat $\mathcal{X}(t)$ un punt en moviment sobre la varietat del grup de Lie $\mathcal{M}$, la seva velocitat $\dot{\mathcal{X}} = \frac{\partial \mathcal{X}}{\partial t}$ en un cert moment t pertany a l'espai tangent a $\mathcal{M}$ al punt $\mathcal{X}$: $\mathcal{T}_{\mathcal{X}}\mathcal{M}$.

Definició 3.4.1: Espai Tangent

Definim l'espai tangent a un punt $\mathcal{X}$ del grup de Lie $\mathcal{M}$ com l'espai generat pels vectors tangents a les corbes regulars de la varietat que passen per $\mathcal{X}$.

La diferenciabilitat del grup de Lie ens dóna la seva unicitat, i l'estructura és la mateixa a tot punt.

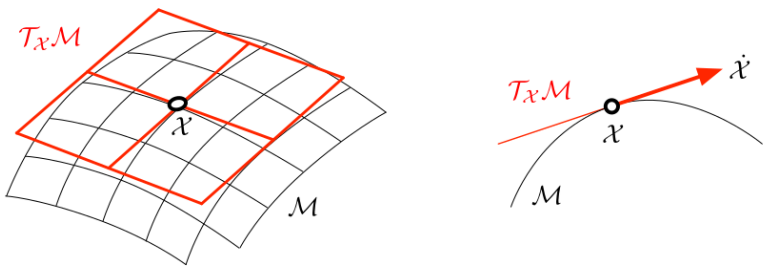

FIGURA 3.1: Una varietat $\mathcal{M}$ i el seu espai tangent $\mathcal{T}_{\mathcal{X}}\mathcal{M}$ al punt $\mathcal{X}$. Cortesia de Joan Solà [Solà et al., 2020]

Definició 3.4.2: L'àlgebra de Lie

En particular, definim l'àlgebra de Lie $\mathfrak{m}$ com l'espai tangent a l'element identitat:

$$\mathfrak{m} = \mathcal{T}_{\mathcal{E}}\mathcal{M}$$

Denotarem els elements de l'espai tangent amb el superíndex "barret" $\boldsymbol{\tau}^\wedge$. Si cal especificar de quin espai tangent parlem el posarem com a superíndex a l'esquerra: ${}^{\mathcal{X}}\boldsymbol{\tau}^\wedge \in \mathcal{T}_{\mathcal{X}}\mathcal{M}$, o bé ${}^{\mathcal{E}}\boldsymbol{\tau}^\wedge \in \mathcal{T}_{\mathcal{E}}\mathcal{M}$. Derivant la condició de grup relacionada amb l'element invers podem trobar una condició necessària i suficient pels elements de l'àlgebra de Lie:

$$\frac{\mathrm{d}(\mathcal{X}^{-1}\mathcal{X})}{\mathrm{d}t} = \frac{\mathrm{d}\mathcal{E}}{\mathrm{d}t} \iff \mathcal{X}^{-1}\dot{\mathcal{X}} + (\dot{\mathcal{X}^{-1}})\mathcal{X} = 0 \tag{3.3}$$

on $(\dot{\mathcal{X}^{-1}})$ és la derivada de l'element invers de $\mathcal{X}$, i entenem $\mathcal{X}^{-1}\dot{\mathcal{X}}$ com l'acció per l'esquerra del grup i $(\dot{\mathcal{X}^{-1}})\mathcal{X}$ l'acció per la dreta.

Definint $\mathbf{v} := \mathcal{X}^{-1}\dot{\mathcal{X}} = -(\dot{\mathcal{X}^{-1}})\mathcal{X}$, aquesta igualtat ens dóna una estructura per l'element $\mathbf{v}$ que s'ha de complir $\forall \mathcal{X} \in \mathcal{M}$. Considerant $\mathcal{X} = \mathcal{E}$ tenim $\mathbf{v} = \dot{\mathcal{X}}$, és a dir, que $\mathbf{v} \in \mathcal{T}_{\mathcal{X}}\mathcal{M}$ quan $\mathcal{X} = \mathcal{E}$. Així doncs, l'estructura de l'àlgebra de Lie és la de l'element $\mathbf{v}$, que denotarem ara com $\mathbf{v}^\wedge = \mathcal{X}^{-1}\dot{\mathcal{X}}$. Com hem dit abans, l'estructura dels espais tangents és la mateixa a tot punt, i així els elements $\dot{\mathcal{X}} \in \mathcal{T}_{\mathcal{X}}\mathcal{M}$ vénen donats per $\dot{\mathcal{X}} = \mathcal{X}\mathbf{v}^\wedge$. Podem trobar un exemple d'aquest procediment a la Fig. 3.2.

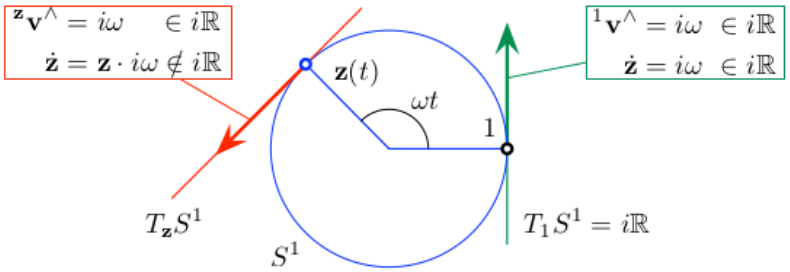

FIGURA 3.2: Sigui $\mathbf{z} \in S^1$ girant a velocitat angular constant ω. La velocitat $\dot{\mathbf{z}}$ pertany als espais tangents $\mathcal{T}_1S^1$ i $\mathcal{T}_{\mathbf{z}}S^1$ quan passa per 1 i $\mathbf{z}$ respectivament. En l'espai tangent $\mathcal{T}_{\mathbf{z}}S^1$ la velocitat és $\dot{\mathbf{z}} = \mathbf{z}i\omega = -\omega \sin \omega t + i\omega \cos \omega t$ expressat en referència global, i ${}^{\mathbf{z}}\mathbf{v}^\wedge = i\omega$ en referència local. Estan relacionades a través de ${}^{\mathbf{z}}\mathbf{v}^\wedge = \mathbf{z}^{-1}\dot{\mathbf{z}} = \mathbf{z}^*\dot{\mathbf{z}}$. En l'espai tangent $\mathcal{T}_1S^1$ aquesta relació és ${}^{1}\mathbf{v}^\wedge = \dot{\mathbf{z}} = i\omega$. Podem veure com clarament l'estructura dels espais tangents ve donada per $i\mathbb{R}$, que és l'àlgebra de Lie. Aquesta és també l'estructura de $\dot{\mathbf{z}}$ a la identitat, raó per la qual identifiquem l'àlgebra de Lie amb aquest espai. Cortesia de Joan Solà [Solà et al., 2020]

3.4.1 Els isomorfismes de l'àlgebra de Lie

Els elements $\boldsymbol{\tau}^\wedge$ de l'espai tangent poden ser expressats com a combinació lineal d'elements d'una base E_i, que anomenem generadors, i són les derivades de $\mathcal{X}$ al voltant de l'origen en la i-èssima direcció. Podem doncs fer servir aquestes coordenades com a vectors de $\mathbb{R}^m$, que notem simplement com $\boldsymbol{\tau}$. Podem passar de $\mathfrak{m}$ a $\mathbb{R}^m$ i viceversa a través dels isomorfismes *hat* $(\cdot)^\wedge$ i *vee* $(\cdot)^\vee$, que són inversos l'un de l'altre:

Definició 3.4.3: Hat i Vee

Definim les funcions "hat" i "vee" com:

$$\begin{aligned}(\cdot)^\wedge \colon \mathbb{R}^m &\to \mathfrak{m} \\ \boldsymbol{\tau} &\mapsto \boldsymbol{\tau}^\wedge := \sum_{i=1}^{m} \tau_i E_i \\ (\cdot)^\vee \colon \mathfrak{m} &\to \mathbb{R}^m \\ \boldsymbol{\tau}^\wedge &\mapsto (\boldsymbol{\tau}^\wedge)^\vee := \boldsymbol{\tau} = \sum_{i=1}^{m} \tau_i \mathbf{e}_i\end{aligned}$$

on els $\mathbf{e}_i$ són la base de $\mathbb{R}^m$ (i $\mathbf{e}_i^\wedge = E_i$).

3.5 L'aplicació exponencial

L'aplicació exponencial exp() és una generalització per grups de Lie de l'exponencial habitual[1], que ens permet passar d'elements de l'àlgebra de Lie a elements del grup. Explicat de manera intuïtiva, l'exponencial plega l'element tangent al voltant de la varietat seguint la geodèsica. L'aplicació inversa és l'aplicació logaritme log(). L'aplicació exp() sorgeix de manera natural considerant que la derivada de $\mathcal{X}$ sobre $\mathcal{M}$ és de la forma:

$$\dot{\mathcal{X}} = \mathcal{X}\mathbf{v}^\wedge$$

que per $\mathbf{v}$ constant és una EDO amb solució

$$\mathcal{X}(t) = \mathcal{X}(0)\exp\left(\mathbf{v}^\wedge t\right)$$

i com tant $\mathcal{X}(0)$ com $\mathcal{X}(t)$ són elements del grup, $\exp(\mathbf{v}^\wedge)$ també ho és. La definició formal per tot grup de Lie és massa complicada per aquest treball, però en el cas d'un grup de Lie matricial coincideix amb l'exponencial matricial. És més, si tenim un grup de Lie multiplicatiu podem identificar l'aplicació exponencial amb la sèrie de Taylor de l'exponencial coneguda.

[1] Que és el cas particular de quan el grup és el grup multiplicatiu del nombres reals positius.

Definició 3.5.1: L'aplicació exponencial

Si definim l'increment tangent com $\boldsymbol{\tau} = \mathbf{v}t \in \mathbb{R}^m$ (de manera que $\boldsymbol{\tau}^\wedge = \mathbf{v}^\wedge t \in \mathfrak{m}$), les aplicacions exponencial i logaritme es poden escriure com:

$$\begin{aligned} \exp\colon \mathfrak{m} &\to \mathcal{M} \\ \boldsymbol{\tau}^\wedge &\mapsto \mathcal{X} = \exp\left(\boldsymbol{\tau}^\wedge\right) \\ \log\colon \mathcal{M} &\to \mathfrak{m} \\ \mathcal{X} &\mapsto \boldsymbol{\tau}^\wedge = \log(\mathcal{X}) \end{aligned}$$

I les expressions tancades surten de fer la serie de Taylor i aprofitar les propietats algebraiques de $\boldsymbol{\tau}^\wedge$ que tingui cada cas.

L'aplicació exponencial té com a propietats importants:

$$\exp\left((t+s)\boldsymbol{\tau}^\wedge\right) = \exp\left(t\boldsymbol{\tau}^\wedge\right)\exp\left(s\boldsymbol{\tau}^\wedge\right) \tag{3.4}$$

$$\exp\left(t\boldsymbol{\tau}^\wedge\right) = \exp\left(\boldsymbol{\tau}^\wedge\right)^t \tag{3.5}$$

$$\exp\left(\mathcal{X}\boldsymbol{\tau}^\wedge\mathcal{X}^{-1}\right) = \mathcal{X}\exp\left(\boldsymbol{\tau}^\wedge\right)\mathcal{X}^{-1} \tag{3.6}$$

on aquesta última es demostra facilment expandint la sèrie de Taylor i simplificant els termes $\mathcal{X}^{-1}\mathcal{X}$

Fem servir també una drecera important, les aplicacions exponencial i logaritme majúscules Exp() i Log(), que ens permeten relacionar elements $\tau \in \mathbb{R}^m$ directament amb elements $\mathcal{X} \in \mathcal{M}$:

Definició 3.5.2: Exponencial majúscula

Les dreceres Exp() i Log() es defineixen de la següent manera:

$$\begin{aligned} \mathrm{Exp}\colon \mathbb{R}^m &\to \mathcal{M} \\ \boldsymbol{\tau} &\mapsto \mathrm{Exp}(\boldsymbol{\tau}) := \exp\left(\boldsymbol{\tau}^\wedge\right) \\ \mathrm{Log}\colon \mathcal{M} &\to \mathbb{R}^m \\ \mathcal{X} &\mapsto \mathrm{Log}(\mathcal{X}) := \log(\mathcal{X})^\vee \end{aligned}$$

3.6 Els operadors suma i resta

Els operadors suma $\oplus$ i resta $\ominus$ ens permeten introduir increments entre elements del grup de Lie (corbada) i expressar-los en el seu espai tangent (pla). Es basen en combinar les aplicacions Exp() i Log() amb una composició. Com la composició del grup no és commutativa en general, definirem els operadors tant per l'esquerra com per la dreta:

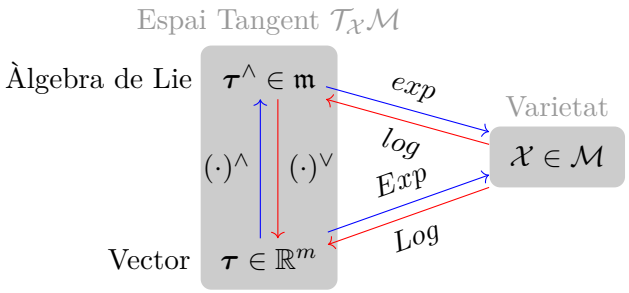

FIGURA 3.3: Diagrama de les aplicacions entre l'espai tangent i el grup de Lie. Hat $(\cdot)^{\wedge}$ i Vee $(\cdot)^{\vee}$ són isomorfismes de representacions de l'espai tangent, $\exp(\cdot)$ i $\log(\cdot)$ relacionen l'àlgebra de Lie amb el grup de Lie, $\mathrm{Exp}(\cdot)$ i $\mathrm{Log}(\cdot)$ són dreceres per poder relacionar directament $\mathbb{R}^m$ i el grup de Lie.

Definició 3.6.1: Operadors $\oplus$ i $\ominus$

Tant $\mathcal{T}_{\mathcal{X}}\mathcal{M}$ com $\mathcal{T}_{\mathcal{E}}\mathcal{M}$ són isomorfs a $\mathbb{R}^m$.

- **Dreta:**

$$\begin{aligned}
\oplus\colon \mathcal{M} \times \mathbb{R}^m &\longrightarrow \mathcal{M} \\
(\mathcal{X}, {}^{\mathcal{X}}\boldsymbol{\tau}) &\longmapsto \mathcal{X} \oplus {}^{\mathcal{X}}\boldsymbol{\tau} := \mathcal{X} \circ \mathrm{Exp}({}^{\mathcal{X}}\boldsymbol{\tau}) = \mathcal{Y} \\
\ominus\colon \mathcal{M} \times \mathcal{M} &\longrightarrow \mathbb{R}^m \\
(\mathcal{Y}, \mathcal{X}) &\longmapsto \mathcal{Y} \ominus \mathcal{X} := \mathrm{Log}(\mathcal{X}^{-1} \circ \mathcal{Y}) = {}^{\mathcal{X}}\boldsymbol{\tau}
\end{aligned}$$

- **Esquerra:**

$$\begin{aligned}
\oplus\colon \mathbb{R}^m \times \mathcal{M} &\longrightarrow \mathcal{M} \\
({}^{\mathcal{E}}\boldsymbol{\tau}, \mathcal{X}) &\longmapsto {}^{\mathcal{E}}\boldsymbol{\tau} \oplus \mathcal{X} := \mathrm{Exp}({}^{\mathcal{E}}\boldsymbol{\tau}) \circ \mathcal{X} = \mathcal{Y} \\
\ominus\colon \mathcal{M} \times \mathcal{M} &\longrightarrow \mathbb{R}^m \\
(\mathcal{Y}, \mathcal{X}) &\longmapsto \mathcal{Y} \ominus \mathcal{X} := \mathrm{Log}(\mathcal{Y} \circ \mathcal{X}^{-1}) = {}^{\mathcal{E}}\boldsymbol{\tau}
\end{aligned}$$

Notem que per l'operador resta la notació és ambigua pels dos casos. En general i a no ser que es digui el contrari, quan treballem amb pertorbacions ho fem en la referència local i per tant estarem fent servir la versió per la dreta de $\oplus$ i $\ominus$.

Per convenció direm que ${}^{\mathcal{X}}\boldsymbol{\tau}$ està expressat en la referència local a $\mathcal{X}$ quan fem servir els operadors per la dreta, i que ${}^{\mathcal{E}}\boldsymbol{\tau}$ està expressat en la referència global quan utilitzem els operadors per l'esquerra.

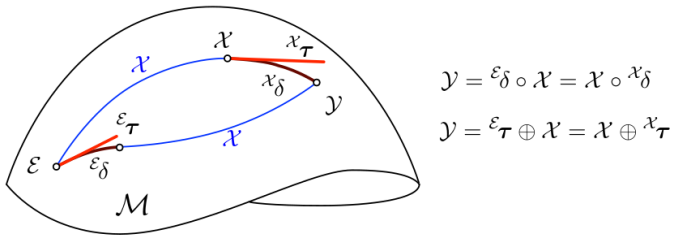

FIGURA 3.4: Es mostren aquí dos camins que relacionen l'origen $\mathcal{E}$ amb el punt $\mathcal{Y}$: $\mathcal{X} \circ {}^{x}\delta$ i ${}^{\varepsilon}\delta \circ \mathcal{X}$. Es diferencien en que ${}^{x}\delta$ està expressat en la referència local i ${}^{\varepsilon}\delta$ en la referència global, i per tant usem l'operador $\oplus$ adient pel cas, dret per al local i esquerre per al global. Cortesia de Joan Solà [Solà et al., 2020]

3.7 L'aplicació adjunta

Identificant $\mathcal{Y}$ en les definicions per la dreta i l'esquerra de $\oplus$ arribem a que ${}^{\varepsilon}\boldsymbol{\tau} \oplus \mathcal{X} = \mathcal{X} \oplus {}^{x}\boldsymbol{\tau}$, és a dir, tenim una relació entre els elements tangents locals i globals. Desenvolupant aquesta relació usant (3.6), arribem a ${}^{\varepsilon}\boldsymbol{\tau}^{\wedge} = \mathcal{X}\,{}^{x}\boldsymbol{\tau}^{\wedge}\mathcal{X}^{-1}$.

Definició 3.7.1: L'aplicació adjunta

Definim l'aplicació *adjunta* de $\mathcal{M}$ en el punt $\mathcal{X}$ com:

$$\mathrm{Ad}_{\mathcal{X}} : \mathfrak{m} \longrightarrow \mathfrak{m} \tag{3.7}$$

$$\boldsymbol{\tau}^{\wedge} \mapsto \mathrm{Ad}_{\mathcal{X}}(\boldsymbol{\tau}^{\wedge}) := \mathcal{X}\boldsymbol{\tau}^{\wedge}\mathcal{X}^{-1} \tag{3.8}$$

de manera que ${}^{\varepsilon}\boldsymbol{\tau} = \mathrm{Ad}_{\mathcal{X}}({}^{x}\boldsymbol{\tau})$.
Aquesta aplicació és l'acció per conjugació del grup sobre la seva àlgebra de Lie. Té dues propietats importants:

$$\text{Linealitat:} \quad \mathrm{Ad}_{\mathcal{X}}(a\boldsymbol{\tau}^{\wedge} + b\boldsymbol{\sigma}^{\wedge}) = a\mathrm{Ad}_{\mathcal{X}}(\boldsymbol{\tau}^{\wedge}) + b\mathrm{Ad}_{\mathcal{X}}(\boldsymbol{\sigma}^{\wedge}) \tag{3.9}$$

$$\text{Homomorfisme:} \quad \mathrm{Ad}_{\mathcal{X}}(\mathrm{Ad}_{\mathcal{Y}}) = \mathrm{Ad}_{\mathcal{X}\mathcal{Y}} \tag{3.10}$$

Gràcies a la linealitat de l'aplicació podem trobar un operador matricial equivalent:

Definició 3.7.2: La matriu adjunta

Definim la matriu *adjunta* associada a l'aplicació adjunta $\mathrm{Ad}_{\mathcal{X}}$ com l'operador matricial

$$\mathbf{Ad}_{\mathcal{X}} : \mathbb{R}^m \longrightarrow \mathbb{R}^m \tag{3.11}$$

$${}^{x}\boldsymbol{\tau} \mapsto \mathbf{Ad}_{\mathcal{X}}\,{}^{x}\boldsymbol{\tau} := {}^{\varepsilon}\boldsymbol{\tau} \tag{3.12}$$

Que es pot calcular a partir d'aplicar $(\cdot)^{\vee}$ a $\mathrm{Ad}_{\mathcal{X}}$:

$$\mathbf{Ad}_{\mathcal{X}}\boldsymbol{\tau} = (\mathcal{X}\boldsymbol{\tau}^{\wedge}\mathcal{X}^{-1})^{\vee} \tag{3.13}$$

Algunes propietats importants de la matriu adjunta són:

$$\mathcal{X} \oplus \boldsymbol{\tau} = (\mathbf{Ad}_{\mathcal{X}}\tau) \oplus \mathcal{X} \tag{3.14}$$

$$\mathbf{Ad}_{\mathcal{X}^{-1}} = \mathbf{Ad}_{\mathcal{X}}^{-1} \tag{3.15}$$

$$\mathbf{Ad}_{\mathcal{X}\mathcal{Y}} = \mathbf{Ad}_{\mathcal{X}}\mathbf{Ad}_{\mathcal{Y}} \tag{3.16}$$

L'utilitzarem per relacionar linealment vectors de l'espai tangent a un punt amb vectors a l'espai tangent a l'origen. Quan parlem de l'*adjunta* ens referirem a aquesta matriu.

3.8 Derivades en grups de Lie

Ja estem en posició per poder definir les derivades en el context dels grups de Lie. Es definiran unes jacobianes sobre varietats que es comportaran com les jacobianes ordinàries a $\mathbb{R}^n$.

Sigui $f : \mathbb{R}^m \to \mathbb{R}^n$ una funció multivariable. La seva Jacobiana és la matriu $n \times m$ formada per les derivades parcials:

$$\mathbf{J} = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_m} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_m} \end{bmatrix} \in \mathbb{R}^{n \times m} \tag{3.17}$$

Cada columna $\mathbf{j}_i$ correspon a

$$\mathbf{j}_i = \frac{\partial f(\mathbf{x})}{\partial x_i} = \lim_{h \to 0} \frac{f(\mathbf{x} + h\mathbf{e}_i) - f(\mathbf{x})}{h} \tag{3.18}$$

on $\mathbf{e}_i$ és l'i-èssim vector de la base de $\mathbb{R}^m$. Per conveniència, usarem la notació més compacta:

$$\mathbf{J} = [\mathbf{j}_1, \ldots, \mathbf{j}_m] =: \lim_{\mathbf{h} \to 0} \frac{f(\mathbf{x} + \mathbf{h}) - f(\mathbf{x})}{\mathbf{h}} \tag{3.19}$$

per referir-nos a ajuntar els $\mathbf{j}_i$ de (3.18) per columnes. Reiterem que només es tracta d'una notació, ja que la divisió per un vector no està definida.

Inspirant-nos en aquesta definició de derivada, prenent els operadors $\oplus$ i $\ominus$ per la dreta, podem definir una derivada per una funció $f : \mathcal{M} \to \mathcal{N}$ sobre varietats substituint les operacions $\{+, -\}$ pels nostres operadors $\{\oplus, \ominus\}$:

$$\frac{Df(\mathcal{X})}{D\mathcal{X}} := \lim_{\boldsymbol{\tau} \to 0} \frac{f(\mathcal{X} \oplus \boldsymbol{\tau}) \ominus f(\mathcal{X})}{\boldsymbol{\tau}} \tag{3.20}$$

que seguint la nostra notació anterior correspon a la matriu que té per columna i-èssima $\lim_{\tau \to 0} \frac{f(\mathcal{X} \oplus \tau\mathbf{e}_i) \ominus f(\mathcal{X})}{\tau}$.

Aquesta nova jacobiana representa la derivada de $f(\mathcal{X})$ respecte de $\mathcal{X}$, expressant les variacions infinitesimals en l'espai tangent. És, per tant, una aplicació lineal entre els espais

tangents locals $\mathcal{T}_{\mathcal{X}}\mathcal{M}$ i $\mathcal{T}_{f(\mathcal{X})}\mathcal{N}$, on les columnes són les derivades direccionals en les direccions de la base.

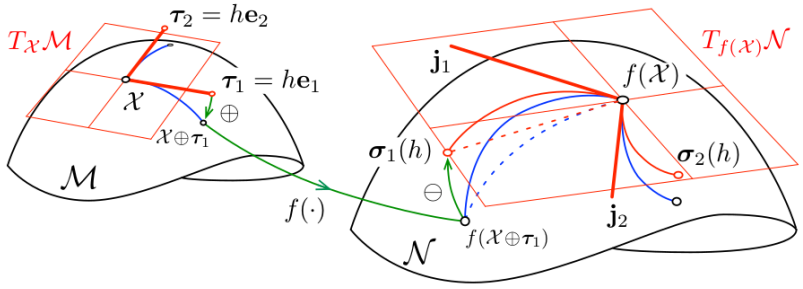

FIGURA 3.5: Jacobiana $\mathbf{J}$ per la dreta d'una funció $f : \mathcal{M} \longrightarrow \mathcal{N}$. Els vectors $\boldsymbol{\tau}_i(h) = h\mathbf{e}_i \in \mathcal{T}_{\mathcal{X}}\mathcal{M}$ en la base canònica són portats als vectors $\boldsymbol{\sigma}_i \in \mathcal{T}_{f(\mathcal{X})}\mathcal{N}$ mitjançant $\oplus$, $f(\cdot)$ i $\ominus$ (fletxes verdes), és a dir $\boldsymbol{\sigma}_i(h) = f(\mathcal{X} \oplus h\mathbf{e}_i) \ominus f(\mathcal{X})$. Les geodèsiques en $\mathcal{M}$ (blau) van a les corbes en blau a $\mathcal{N}$ (podrien no ser geodèsiques), que portem a l'espai tangent (corba vermella fina). Les columnes $\mathbf{j}_i$ de $\mathbf{J}$ corresponen al límit $\mathbf{j}_i = \lim_{h\to 0} \frac{\boldsymbol{\sigma}_i(h)}{h}$ i tenim així que $\mathbf{J}$ ens porta de manera lineal vectors de $\mathcal{T}_{\mathcal{X}}\mathcal{M} \cong \mathbb{R}^m$ a vectors de $\mathcal{T}_{f(\mathcal{X})}\mathcal{N} \cong \mathbb{R}^n$. Cortesia de Joan Solà [Solà et al., 2020]

Per valors petits de $\boldsymbol{\tau}$ la següent aproximació és vàlida:

$$f(\mathcal{X} \oplus \boldsymbol{\tau}) \underset{\boldsymbol{\tau}\to 0}{\longrightarrow} f(\mathcal{X}) \oplus \frac{Df(\mathcal{X})}{D\mathcal{X}}\boldsymbol{\tau} \in \mathcal{N} \tag{3.21}$$

Podem fer el mateix tipus de definició per trobar jacobianes per l'esquerra (i.e. que fan servir els operadors $\oplus$ i $\ominus$ per l'esquerra) i jacobianes creuades que els barregen. Com en aquest treball no les fem servir i el procediment és anàleg, no entrarem en més detall. Sí que mencionarem que les jacobianes per la dreta i l'esquerra estan relacionades a través de l'adjunta:

$$\frac{^{\mathcal{E}}Df(\mathcal{X})}{D\mathcal{X}}\mathbf{Ad}_{\mathcal{X}} = \mathbf{Ad}_{f(\mathcal{X})}\frac{^{\mathcal{X}}Df(\mathcal{X})}{D\mathcal{X}} \tag{3.22}$$

D'ara endavant, com a notació es farà servir $\mathbf{J}^{\mathcal{Y}}_{\mathcal{X}} := \frac{D\mathcal{Y}}{D\mathcal{X}}$. També anomenarem *jacobiana per la dreta de la varietat* $\mathcal{M}$ a la jacobiana de $\mathcal{X} = \mathrm{Exp}(\boldsymbol{\tau})$:

$$\mathbf{J}_r(\boldsymbol{\tau}) := \frac{D\,\mathrm{Exp}(\boldsymbol{\tau})}{D\boldsymbol{\tau}} \tag{3.23}$$

La demostració que aquestes jacobianes compleixen la regla de la cadena és a l'apèndix. A, on detallem també algunes regles de derivació útils.

3.9 Incertesa en varietats, propagació de covariancies

Definim pertorbacions locals $\boldsymbol{\tau}$ al voltant d'un punt $\hat{\mathcal{X}} \in \mathcal{M}$ a l'espai tangent $\mathcal{T}_{\hat{\mathcal{X}}}\mathcal{M}$, usant els operadors $\oplus$ i $\ominus$ per la dreta:

$$\mathcal{X} = \hat{\mathcal{X}} \oplus \boldsymbol{\tau} \tag{3.24}$$

$$\boldsymbol{\tau} = \mathcal{X} \ominus \hat{\mathcal{X}} \tag{3.25}$$

Definició 3.9.1: Matriu de covariàncies

Podem definir de manera natural matrius de covariàncies $\Sigma_{\mathcal{X}}$ sobre l'espai tangent a $\hat{\mathcal{X}}$ mitjançant l'operador esperança $\mathbb{E}[\cdot]$:

$$\Sigma_{\mathcal{X}} := \mathbb{E}[(\boldsymbol{\tau} - \mathbb{E}[\boldsymbol{\tau}])(\boldsymbol{\tau} - \mathbb{E}[\boldsymbol{\tau}])^\top] = \mathbb{E}[\boldsymbol{\tau}\boldsymbol{\tau}^\top] = \mathbb{E}[(\mathcal{X} \ominus \hat{\mathcal{X}})(\mathcal{X} \ominus \hat{\mathcal{X}})^\top] \in \mathbb{R}^{m \times m} \tag{3.26}$$

on m és la dimensió de l'espai tangent i hem usat que $\mathbb{E}[\boldsymbol{\tau}] = 0$. Això ens permet definir variables aleatòries normals en varietats: $\mathcal{X} \sim \mathcal{N}(\hat{\mathcal{X}}, \Sigma_{\mathcal{X}})$. Notem que tot i que escrivim $\Sigma_{\mathcal{X}}$, la covariància correspon en realitat a la pertorbació tangencial $\boldsymbol{\tau}$.

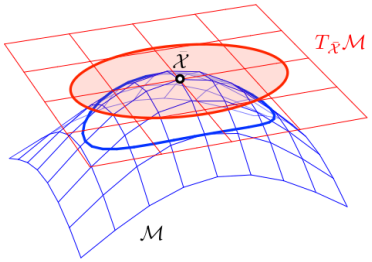

FIGURA 3.6: La incertesa al voltant de $\hat{\mathcal{X}}$ s'expressa com una covariància en l'espai tangent al punt (el·lipse vermella). Usant l'operador $\oplus$ portem l'el·lipse de probabilitat a sobre del grup de Lie. Cortesia de Joan Solà [Solà et al., 2020]

La propagació d'una matriu de covariàncies a través d'una funció $f : \mathcal{M} \to \mathcal{N}$, ve donada per

$$\Sigma_{f(\mathcal{X})} \approx \frac{Df}{D\mathcal{X}} \Sigma_{\mathcal{X}} \frac{Df}{D\mathcal{X}}^\top \tag{3.27}$$

on hem usat l'aproximació (3.21) amb la definició de la jacobiana (3.20).

Hem definit aquí la propagació per pertorbacions expressades en referència local. També es poden definir en referència global fent servir els operadors per l'esquerra. Com els vectors en referència global i local estan relacionats a través de l'adjunta, podrem usar-la per passar de covariàncies en referència global a covariàncies en referència local:

$${}^{\mathcal{E}}\Sigma_{\mathcal{X}} = \mathbf{Ad}_{\mathcal{X}}{}^{\mathcal{X}}\Sigma_{\mathcal{X}}\mathbf{Ad}_{\mathcal{X}}{}^\top \tag{3.28}$$

4 El sensor IMU i el seu grup de Lie

4.1 IMUs

Una IMU (**I**nertial **M**easurement **U**nit) és un sensor que mesura acceleració lineal i velocitats angulars en una referència inercial. Típicament, això s'aconsegueix combinant tres acceleròmetres i tres giroscopis de manera ortogonal dos a dos.

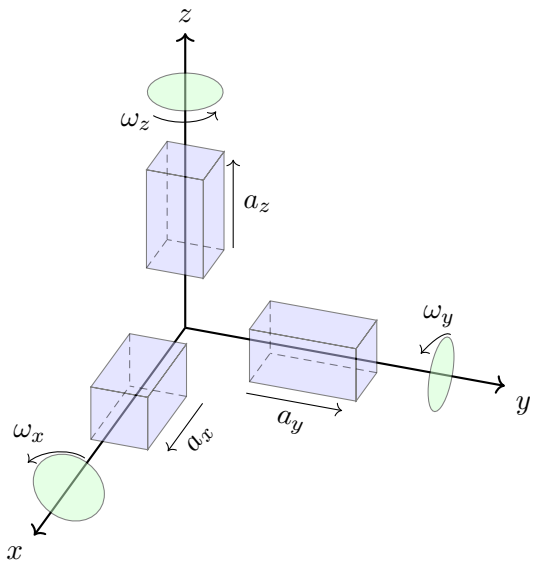

FIGURA 4.1: Diagrama de la configuració d'una IMU. Els blocs blaus són els acceleròmetres, que mesuren les acceleracions lineals en cada eix a_i. Els discs verds són els giroscopis que mesuren velocitats angulars en cada eix ω_i

.

Definició 4.1.1: Acceleròmetre

Un acceleròmetre és un sensor que mesura l'acceleració que experimenta respecte un observador inercial en repòs.
Una conseqüència d'això és que un acceleròmetre en repòs a la terra mesurarà una acceleració de magnitud g cap amunt. En canvi en condicions de caiguda lliure un acceleròmetre marca 0.

Definició 4.1.2: Giroscopi

Un giroscopi és un sensor que mesura la orientació en què està, i la velocitat angular com a extensió d'això quan es combina amb un rellotge.

Les fonts d'error en les dades retornades per una IMU són diverses:

1. **Alineació:** En una IMU ideal, els eixos sobre els quals estan muntats els acceleròmetres són ortogonals i estan alineats amb els eixos dels giroscopis. En el món real, a causa d'imprecisions durant la construcció, això no es dóna i indueix errors d'alineació.
2. **Escala:** En alguns casos, les dades que retorna la IMU poden presentar un error lineal, causant que les dades retornades siguin proporcionals a les reals. Aquests efectes són més aparents en valors alts d'acceleració lineal i velocitat de gir.
3. **Biaix:** Per culpa d'imprecisions durant el procés de construcció i efectes de la temperatura sobre els materials, cada eix de la IMU està separat de l'origen per una magnitud independent dels altres eixos. Això causa que cadascun dels termes retornats estigui afectat per un terme additiu desconegut. Ho modelem mitjançant 2 vectors de 3 components $\mathbf{a}_b$ i $\boldsymbol{\omega}_b$
4. **Soroll:** Com tots els sensors, la IMU també està afectada pel soroll de mesura. El soroll a cada eix és independent, així que tenim també 2 vectors de 3 components $\mathbf{a}_n$ i $\boldsymbol{\omega}_n$, que modelem amb una distribució normal: $\mathbf{a}_n \sim \mathcal{N}(0, \Sigma_{a,n}), \boldsymbol{\omega}_n \sim \mathcal{N}(0, \Sigma_{\omega,n})$, on $\Sigma_{a,n}$ i $\Sigma_{\omega,n}$ són les matrius de covariància i depenen del sensor.

Aquestes fonts d'error no tenen el mateix impacte en les mesures en condicions experimentals. Els errors d'escala i d'alineació solen ser negligibles, mentre que els errors causats pel biaix són els més pronunciats, tant per la dificultat d'estimar-los com per la integració d'aquests durant l'estimació de posició i velocitat.

El model d'error del sensor que utilitzarem serà el següent:

$$\begin{aligned} \mathbf{a}_{mesura} &= \mathbf{a}_{real} + \mathbf{a}_b + \mathbf{a}_n \\ \boldsymbol{\omega}_{mesura} &= \boldsymbol{\omega}_{real} + \boldsymbol{\omega}_b + \boldsymbol{\omega}_n \end{aligned} \tag{4.1}$$

Les IMU són de gran interès en la robòtica, ja que en teoria es pot reconstruir la trajectòria del robot a través d'integrar respecte el temps les seves dades. També permeten detectar situacions particulars, com per exemple que el robot rellisqui.

4.2 IMU i Graph-SLAM

Les dades de la IMU en un graf de factors estaran localitzades en factors entre estats del robot consecutius. La freqüència d'aquests sensors és molt alta, es poden generar milers de mesures en un únic segon. Si simplement afegíssim un node d'estat després de cada dada rebuda, el nostre graf creixeria de manera molt ràpida i es tornaria intractable. Voldríem, doncs, tenir una manera d'integrar totes les mesures de la IMU entre dos estats "interessants",

generant així un factor de moviment entre ells. Això ens permetrà mantenir el problema d'una grandària tractable.

Definició 4.2.1: Keyframe

Un keyframe és un node d'estat del graf "interessant". Això pot significar diferents coses segons el que representi el node, i el criteri per decidir si és o no un keyframe es pot definir lliurement.
En comptes de fer un graf de factors on cada mesura uneix dos estats, construirem un graf de keyframes on els factors entre ells aglutinaran totes les mesures que les relacionen.
L'exemple més senzill seria crear un keyframe cada segon. En aquest cas s'afegiria un node d'estat al graf cada vegada que passa un segon. Un exemple més complex seria, en un robot bípede, crear un keyframe cada vegada que un dels peus toca el terra. Aleshores, cada node d'estat del graf correspondria a un pas del robot.

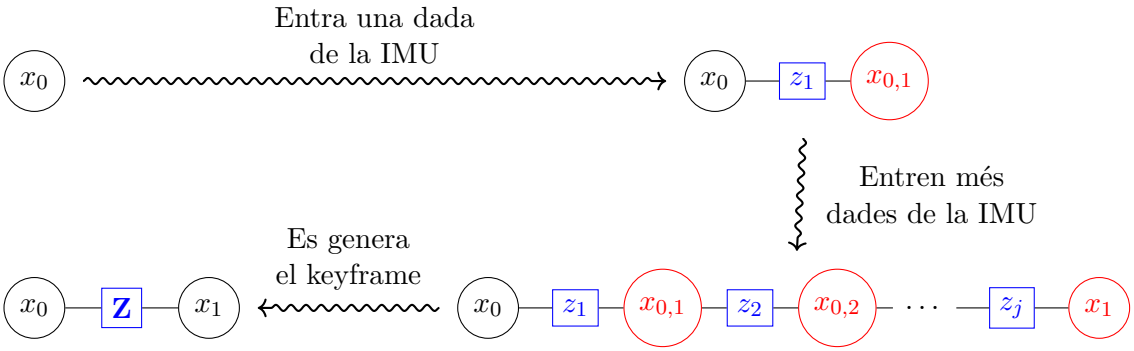

FIGURA 4.2: Generació de Keyframe en el cas de la IMU. Cada vegada que rebem una dada z_i de la IMU tindrem un nou estat relacionat amb l'anterior a través de z_i. En el moment que es decideix crear un keyframe s'aglutinen totes les dades z_i en un únic factor **Z** que relaciona l'anterior keyframe amb aquest, evitant el creixement descontrolat del graf. A la pràctica no es generaran els nodes vermells i simplement anirem aglutinant les z_i fins que es generi el keyframe.

Integrant les equacions del moviment, veiem que depenen fortament de:

- La posició, orientació i velocitat inicials (que haurem d'estimar).
- Les velocitats angulars i acceleracions lineals (que són les dades del sensor).
- Els biaixos de la IMU (que haurem d'estimar).

Atès que petits canvis en les estimacions de l'estat inicial i els biaixos de la IMU afecten a tota la integració del moviment, necessitem una manera d'evitar haver de tornar a integrar totes les dades a cada iteració de l'optimització.

Introduïm aquí el concepte de preintegració. Definirem unes noves magnituds, les *deltes*, que depenen exclusivament de les dades de la IMU i dels seus biaixos. D'aquesta manera, trobarem una expressió per l'estat que depengui d'una banda d'aquestes deltes, i d'altra

banda de l'estat inicial. Això ens permetrà recalcular l'estat quan canvii l'estimació de l'estat inicial sense haver de reintegrar totes les dades.

Siguin uns estats $\mathbf{x}_i$ i $\mathbf{x}_j$. Recordant els conceptes introduïts al capítol 2, aquests estats estan relacionats per un factor que resumeix el moviment entre ells, representat en aquest cas per una delta Δ_{ij}. Aquesta relació es nota amb un operador $\boxplus$ de manera que

$$\mathbf{x}_j = \mathbf{x}_i \boxplus \Delta_{ij} \tag{4.2}$$

i considerem també l'operador $\boxminus$ de manera que

$$\Delta_{ij} = \mathbf{x}_j \boxminus \mathbf{x}_i \tag{4.3}$$

Remarquem que no estem definint la delta a través d'aquest operador, la delta ve de les dades (i el biaix).

Siguin $\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j$ estimacions dels estats anteriors. Per (4.3) podem trobar una $\hat{\Delta}_{ij}$ que relaciona aquestes estimacions. Podem definir una funció d'error $\mathbf{e}_{ij}(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j)$

$$\mathbf{e}_{ij} = \Delta_{ij} \ominus \hat{\Delta}_{ij} = \Delta_{ij} \ominus (\hat{\mathbf{x}}_j \boxminus \hat{\mathbf{x}}_i) \in \mathbb{R}^n \tag{4.4}$$

on $\ominus$ és l'operador vist al capítol 3.[1]

Voldrem poder construir la Δ_{ij} incrementalment a la mateixa freqüència a què arriben les dades de la IMU. És a dir, per cada dada de la IMU que arriba (i que ens relaciona $\mathbf{x}_j$ i $\mathbf{x}_k$ estats consecutius), generar una delta δ_k tal que:

$$\Delta_{ik} = \Delta_{ij} \boxplus \delta_k \tag{4.5}$$

i així tenir $\mathbf{x}_k = \mathbf{x}_j \boxplus \delta_k = \mathbf{x}_i \boxplus \Delta_{ij} \boxplus \delta_k = \mathbf{x}_i \boxplus \Delta_{ik}$.

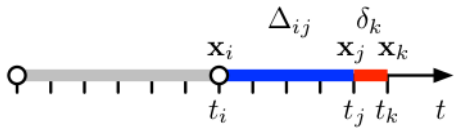

FIGURA 4.3: La delta preintegrada Δ_{ij} conté tot el moviment des de el key-frame $\mathbf{x}_i$ fins a temps t_j, mentre que la delta actual δ_k conté el moviment des de $\mathbf{x}_j$ fins a $\mathbf{x}_k$, on $t_k - t_j$ és el temps de mostreig de la IMU. Cortesia de Dinesh Atchuthan [Atchuthan, 2018]

D'aquesta manera:

$$\mathbf{x}_2 = \mathbf{x}_1 \boxplus \Delta_{1,2} = \mathbf{x}_0 \boxplus (\Delta_{0,1} \boxplus \Delta_{1,2}) = \mathbf{x}_0 \boxplus \Delta_{0,2} = \mathbf{x}_0 \boxplus \left(\sum_{\delta_k \in (\mathbf{x}_0, \mathbf{x}_2)}^{\boxplus} \delta_k \right) \tag{4.6}$$

tenim una expressió que només depèn de l'estat inicial i les deltes, que depenen només de les dades de la IMU, tal com volíem.

[1] Veurem més endavant que és un grup de Lie.

Com hem dit abans, però, les dades de la IMU estan afectades per un biaix que haurem d'anar estimant. El canvi, en l'estimació del biaix sí afecta a les deltes, i per tant haurem de trobar una manera d'actualitzar-les sense haver de tornar a integrar-ho tot. Això ho aconseguirem mitjançant un procés de linealització de l'efecte del biaix sobre les deltes, permetent-nos corregir les deltes a posteriori mitjançant unes jacobianes precalculades.

Així, explicitant la dependència en el biaix del sensor $\mathbf{c}_i$, l'equació (4.5) passa a ser:

$$\bar{\Delta}_{ik}(\bar{\mathbf{c}}_i) = \bar{\Delta}_{ij}(\bar{\mathbf{c}}_i) \boxplus \delta_k(\bar{\mathbf{c}}_i) \tag{4.7}$$

On posem la barra per denotar que són les deltes i els biaixos precorrecció. Finalment, donada una nova estimació $\mathbf{c}_i$ del biaix, podrem corregir la delta a posteriori (en el nostre cas al calcular els errors per minimitzar la funció de cost) utilitzant una jacobiana:

$$\Delta_{ij} = \bar{\Delta}_{ij}(\bar{\mathbf{c}}_i) \oplus \mathbf{J}_{\mathbf{c}_i}^{\Delta_{ij}}(\mathbf{c}_i - \bar{\mathbf{c}}_i) \tag{4.8}$$

on $\mathbf{J}_{\mathbf{c}_i}^{\Delta_{ij}}$ és la jacobiana de la delta respecte el biaix. És a dir, corregim utilitzant l'aproximació de primer ordre.

Introduirem a partir d'aquí les definicions formals de tots aquests objectes.

4.3 Preintregració de les dades.

Al llarg de la resta d'aquest capítol donarem les derivacions de les deltes i explicarem el procediment per construir la teoria de preintegració, que ens permetrà fer us de la IMU de manera eficient en els nostres problemes de SLAM. Tot i que aquest treball està centrat en la preintegració de la IMU per a problemes bidimensionals on el robot es mou sobre un pla no inclinat, explicarem primer el procediment sobre l'espai tridimensional, que ens permetrà donar un sentit físic a les nostres deltes. A més a més, fora del problema teòric el nostre robot sí que viu en un espai tridimensional, i les IMUs estan dissenyades amb 3 eixos en ment.[2] El canvi posterior al cas 2D és senzill i molt semblant al cas 3D. El farem abans de començar a aplicar teoria de Lie, ja que és on els dos casos comencen a divergir fortament. La descripció completa del cas 3D pot ser trobada a [Atchuthan, 2018].

4.3.1 Integració de l'estat en referència absoluta

Definirem l'estat de la IMU (és equivalent a l'estat del robot, posat que hi està fixada) com $\mathbf{x} = (\mathbf{p}, \mathbf{v}, \mathbf{R})$, on $\mathbf{p}$ és la posició, $\mathbf{v}$ és la velocitat (tots dos en referència globals) i $\mathbf{R}$ és la orientació en forma de matriu de rotació. En particular, $\mathbf{R}$ és la matriu que ens porta un vector en referència local a la referència global. Usarem l'operador antisimètric $[\cdot]_\times$, que per

[2] De fet es podria construir una IMU específicament pel cas 2D fent servir dos acceleròmetres posicionats sobre els eixos X i Y, i un giroscopi sobre l'eix Z. Una IMU normal consta de 3 acceleròmetres i 3 giroscopis posicionats sobre cadascun dels 3 eixos.

vectors de $\mathbb{R}^3$ és

$$[\boldsymbol{\omega}]_\times := \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \in \mathbb{R}^{3\times 3} \tag{4.9}$$

L'evolució de $\mathbf{x}$ està governada per les equacions cinemàtiques següents:

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{g} + \mathbf{R}\mathbf{a} \\ \dot{\mathbf{R}} &= \mathbf{R}\,[\boldsymbol{\omega}]_\times \end{aligned} \tag{4.10}$$

on $\mathbf{g}$ és el vector de la gravetat i definim $\mathbf{b} = (\mathbf{a}, \boldsymbol{\omega})$ com el vector de *magnituds inercials*: les magnituds de l'acceleració lineal i la velocitat angular experimentades per la IMU en la seva referència local. Aquestes magnituds són obtingudes en temps discrets t_j a partir de mesures de la IMU amb soroll i biaix, és a dir:

$$\begin{aligned} \mathbf{a}_j &= \mathbf{a}_{m,j} - \mathbf{a}_{b,j} - \mathbf{a}_n \\ \boldsymbol{\omega}_j &= \boldsymbol{\omega}_{m,j} - \boldsymbol{\omega}_{b,j} - \boldsymbol{\omega}_n \end{aligned} \tag{4.11}$$

on hem fet servir el model del sensor descrit a (4.1): $\bullet_m$ són les dades mesurades, $\bullet_b$ són els biaixos i $\bullet_n$ el soroll. Suposant que aquestes magnituds són constants durant el període de mostreig de la IMU $\delta t = t_k - t_j$, tenim la discretització de les equacions del moviment següent:

$$\begin{aligned} \mathbf{p}_k &= \mathbf{p}_j + \mathbf{v}_j \delta t + \frac{1}{2}\mathbf{g}\delta t^2 + \frac{1}{2}\mathbf{R}_j\mathbf{a}_j\delta t^2 \\ \mathbf{v}_k &= \mathbf{v}_j + \mathbf{g}\delta t + \mathbf{R}_j\mathbf{a}_j\delta t \\ \mathbf{R}_k &= \mathbf{R}_j \exp\big([\boldsymbol{\omega}_j\delta t]_\times\big) \end{aligned} \tag{4.12}$$

Aquesta suposició és molt important, com veurem més endavant. A més a més, la podem fer gràcies a les altes freqüències a les quals treballa la IMU, sobretot en situacions poc dinàmiques.

4.3.2 Definicions de les deltes

Considerem una referència en caiguda lliure sense rotació $\mathcal{G}_t$ en un espai 3D. Una IMU ideal (sense biaixos ni sorolls) sobre aquesta referència mesuraria 0 tant en acceleració lineal com en velocitat angular. Mesures no nul·les correspondrien a moviment relatiu entre la IMU i $\mathcal{G}_t$. Per tant, podem identificar les deltes de la IMU com els increments de moviment respecte una referència en caiguda lliure sense rotació. En altres paraules: la posició, velocitat i orientació de la IMU expressades en la referència $\mathcal{G}_t$.

En un moment t_i, inicialitzem $\mathcal{G}_i$ a $\mathbf{x}_i = (\mathbf{p}_i, \mathbf{v}_i, \mathbf{R}_i)$, l'estat de la IMU en el moment t_i. En un moment t_j posterior, la referència $\mathcal{G}_j$ s'ha mogut sota l'acció de la gravetat $\mathbf{g}$ i l'estat de la IMU és $\mathbf{x}_j = (\mathbf{p}_j, \mathbf{v}_j, \mathbf{R}_j)$. La variació en l'estat Δ_{ij} queda definida com la variació en posició, velocitat i orientació de la IMU entre $\mathcal{G}_i$ i $\mathcal{G}_j$:

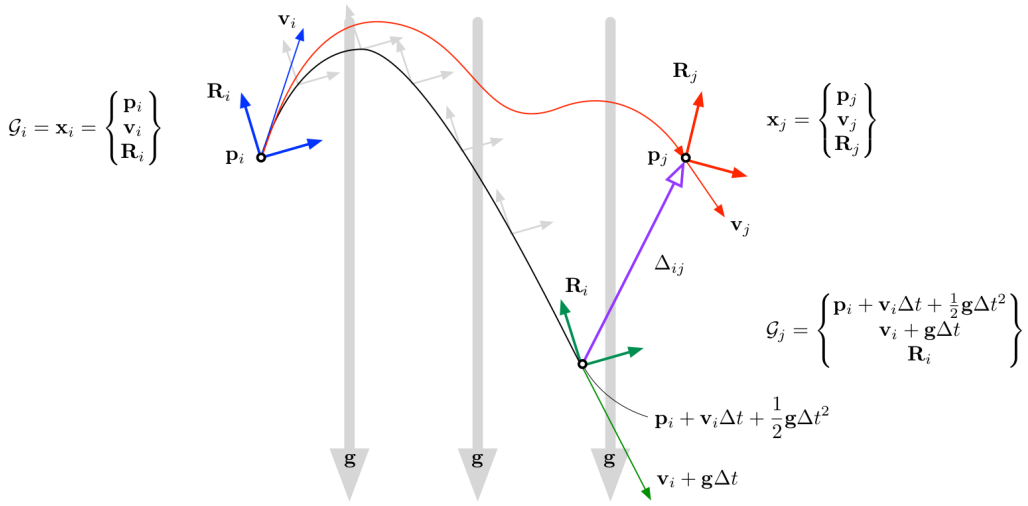

FIGURA 4.4: La referència sense gir en caiguda lliure $\mathcal{G}_t$ segueix una trajectòria parabòlica governada només per la gravetat $\mathbf{g}$ i determinada per les condicions inicials $\mathbf{p}_i$, $\mathbf{v}_i$ i $\mathbf{R}_i$. La delta entre temps i i j es defineix com l'estat de la IMU $\mathbf{x}_j$ en temps j expressat en la referència $\mathcal{G}_j$. Cortesia de Joan Solà [Solà, 2019]

Definició 4.3.1: Δ_{ij} en 3D

A partir de l'explicació de com interpretem les deltes, definim Δ_{ij} a través de les seves parts:

$$\begin{aligned} \Delta\mathbf{p}_{ij} &= \mathbf{R}_i^\top \left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2 \right) \\ \Delta\mathbf{v}_{ij} &= \mathbf{R}_i^\top \left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij} \right) \\ \Delta\mathbf{R}_{ij} &= \mathbf{R}_i^\top \mathbf{R}_j \end{aligned} \tag{4.13}$$

on $\Delta t_{ij} := t_j - t_i$. Aquesta definició per les Δ_{ij} coincideix amb la donada per [Lupton and Sukkarieh, 2009; Forster et al., 2015] però li hem donat aquí una interpretació física.

En tot moment podrem recuperar l'estimació de l'estat $\mathbf{x}_j$ a partir de l'estimació de l'estat $\mathbf{x}_i$ i la delta del moviment Δ_{ij}:

$$\begin{aligned} \mathbf{p}_j &= \mathbf{p}_i + \mathbf{v}_i\Delta t_{ij} + \frac{1}{2}\mathbf{g}\Delta t_{ij}^2 + \mathbf{R}_i\Delta\mathbf{p}_{ij} \\ \mathbf{v}_j &= \mathbf{v}_i + \mathbf{g}\Delta t_{ij} + \mathbf{R}_i\Delta\mathbf{v}_{ij} \\ \mathbf{R}_j &= \mathbf{R}_i\Delta\mathbf{R}_{ij} \end{aligned} \tag{4.14}$$

Notem que aquestes deltes formen un grup:

Definició 4.3.2: Composició de les deltes

La composició $\Delta_{ik} := \Delta_{ij} \boxplus \Delta_{jk}$ està definida per

$$\begin{aligned} \Delta\mathbf{p}_{ik} &= \Delta\mathbf{p}_{ij} + \Delta\mathbf{v}_{ij}\Delta t_{jk} + \Delta\mathbf{R}_{ij}\Delta\mathbf{p}_{jk} \\ \Delta\mathbf{v}_{ik} &= \Delta\mathbf{v}_{ij} + \Delta\mathbf{R}_{ij}\Delta\mathbf{v}_{jk} \\ \Delta\mathbf{R}_{ik} &= \Delta\mathbf{R}_{ij}\Delta\mathbf{R}_{jk} \end{aligned} \tag{4.15}$$

Amb element identitat $\Delta_0 = [(0,0,0),(0,0,0),I_3]$ i inversa Δ^{-1} definida per:

$$\begin{aligned} \Delta\mathbf{p}^{-1} &= -\Delta\mathbf{R}^\top(\Delta\mathbf{p} - \Delta\mathbf{v}\Delta t) \\ \Delta\mathbf{v}^{-1} &= -\Delta\mathbf{R}^\top\Delta\mathbf{v} \\ \Delta\mathbf{R}^{-1} &= \Delta\mathbf{R}^\top \end{aligned} \tag{4.16}$$

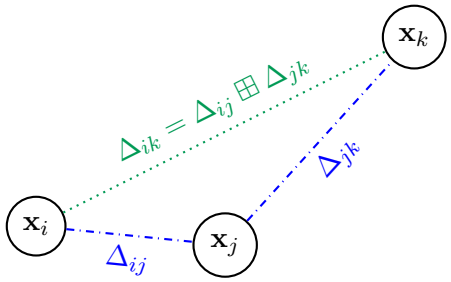

FIGURA 4.5: Composició de les deltes

4.3.3 Preintegració incremental de les deltes

Substituint la integració discreta de (4.12) en la definició de les deltes (4.13) arribem a l'expressió que ens permet actualitzar la delta de manera incremental:

$$\begin{aligned} \Delta\mathbf{p}_{ik} &= \Delta\mathbf{p}_{ij} + \Delta\mathbf{v}_{ij}\delta t + \frac{1}{2}\Delta\mathbf{R}_{ij}\mathbf{a}_j\delta t^2 \\ \Delta\mathbf{v}_{ik} &= \Delta\mathbf{v}_{ij} + \Delta\mathbf{R}_{ij}\mathbf{a}_j\delta t \\ \Delta\mathbf{R}_{ik} &= \Delta\mathbf{R}_{ij}\exp\big([\boldsymbol{\omega}_j\delta t]_\times\big) \end{aligned} \tag{4.17}$$

Aquestes equacions són anàlogues a les del moviment d'un cos en una referència inercial sota acceleració i velocitat de gir constants. Per un desenvolupament més detallat, veure l'apèndix. B.1.

Fem notar que aquestes expressions són dependents només d'elements de la delta Δ_{ij} i de les magnituds inercials. Definirem ara una nova delta δ_{jk} a partir d'aquestes magnituds en el moment t_j:

$$\begin{aligned} \delta\mathbf{p}_{jk} &= \frac{1}{2}\mathbf{a}_j\delta t^2 \\ \delta\mathbf{v}_{jk} &= \mathbf{a}_j\delta t \\ \delta\mathbf{R}_{jk} &= \mathbf{R}\exp\big([\boldsymbol{\omega}\delta t]_\times\big) \end{aligned} \tag{4.18}$$

i així (4.17) és la composició $\Delta_{ik} = \Delta_{ij} \boxplus \delta_{jk}$ descrita abans. Veiem com clarament podrem anar integrant les dades de la IMU segons arriben en una única delta que ens relacionarà els keyframes consecutius. Com a notació, farem servir Δ per les deltes preintegrades i δ per la delta actual, corresponent a les últimes dades de la IMU.

La delta actual definida aquí és la mateixa que surt a altres articles que parlen de integració de IMUs com [Forster et al., 2015]. Veurem, però, que fent servir la teoria de Lie arribarem a una expressió millor.

4.3.4 El cas 2D

Hem descrit fins ara la base física tridimensional sobre la qual hem definit les deltes. En aquesta secció explicarem quines coses canvien al passar al cas 2D, i les que no es mencionin seran iguals que abans.

Per passar al cas 2D projectarem sobre el pla XY i ens desfarem així de la gravetat $\mathbf{g}$. L'estat $\mathbf{x}$ continua estant definit com $\mathbf{x} = (\mathbf{p}, \mathbf{v}, \mathbf{R})$, però ara $\mathbf{p}$ i $\mathbf{v}$ són vectors 2D corresponents a les components x i y del cas 3D. $\mathbf{R}$ és la matriu de rotació corresponent al gir al voltant de l'eix z del cas 3D. L'operador antisimètric $[\cdot]_\times$ en 2D és

$$[\omega]_\times = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} \tag{4.19}$$

Les equacions cinemàtiques (4.10) que governaven l'evolució de l'estat passen a ser:

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{R}\mathbf{a} \\ \dot{\mathbf{R}} &= \mathbf{R}\,[\omega]_\times \end{aligned} \tag{4.20}$$

on simplement ha desaparegut la dependència de la gravetat. La integració discreta (4.12) passa a ser ara:

$$\begin{aligned} \mathbf{p}_k &= \mathbf{p}_j + \mathbf{v}_j \delta t + \frac{1}{2}\mathbf{R}_j \mathbf{a}_j \delta t^2 \\ \mathbf{v}_k &= \mathbf{v}_j + \mathbf{R}_j \mathbf{a}_j \delta t \\ \mathbf{R}_k &= \mathbf{R}_j \exp\big([\omega_j \delta t]_\times\big) \end{aligned} \tag{4.21}$$

amb $\delta t := t_k - t_j$ com abans.

La definició de les deltes al cas 2D canvia només en el fet que ara la referència es mou a velocitat constant sense rotació, ja que l'única acceleració present era la de $\mathbf{g}$:

Definició 4.3.3: Δ_{ij} en 2D

$$\begin{aligned} \Delta\mathbf{p}_{ij} &= \mathbf{R}_i^\top (\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i \Delta t_{ij}) \\ \Delta\mathbf{v}_{ij} &= \mathbf{R}_i^\top (\mathbf{v}_j - \mathbf{v}_i) \\ \Delta\mathbf{R}_{ij} &= \mathbf{R}_i^\top \mathbf{R}_j \end{aligned} \tag{4.22}$$

on $\Delta t_{ij} := t_j - t_i$ com abans. La composició és exactament com abans tenint en compte els canvis de dimensió.

És fàcil comprovar que substituir (4.21) a (4.22) ens porta a la mateixa expressió per la preintegració incremental que al cas 3D.

4.4 El grup de Lie de la IMU

Fins ara, a causa de com està descrita en la literatura la definició de les deltes, totes les integracions i definicions han sigut per blocs. Per definir el grup de Lie de la IMU tenim dues opcions, la versió per blocs que tracta cada component de la delta per separat amb la seva pròpia varietat (i.e. les parts no interactuen entre si i les podem tractar una a una), o la versió compacta on considerem les deltes membres d'una única varietat (i.e. les parts interactuen i es defineixen les operacions sobre el conjunt total). Aquesta segona versió és la que és consistent amb la teoria de Lie definida al capítol 3 (i és més precisa a la pràctica) i és la que desenvoluparem en aquesta secció, però hem de mencionar que a l'hora d'implementar-ho en codi ens hem vist obligats a usar les definicions de les Jacobianes corresponents al grup de Lie per blocs, a causa de no haver trobat l'expressió tancada d'una de les jacobianes del grup compacte. No obstant, sí que s'ha pogut implementar sense problema la preintegració com a grup de Lie compacte.

4.4.1 Elements del grup de Lie

Podem representar les deltes de la IMU de les següents maneres equivalents:

$$\Delta = \begin{bmatrix} \Delta\mathbf{R} & \Delta\mathbf{v} & \Delta\mathbf{p} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \Delta\mathbf{p} \\ \Delta\mathbf{v} \\ \Delta\mathbf{R} \\ \Delta t \end{bmatrix} \tag{4.23}$$

Observem ara que si afegim Δt al grup (amb la composició suma, identitat 0 i inversa $-\Delta t$), la versió 2D de la composició de grup que havíem definit a (4.15), coincideix amb el producte de matrius quan representem les deltes en la forma matricial:

$$\Delta_i \Delta_j = \begin{bmatrix} \Delta\mathbf{R}_i\Delta\mathbf{R}_j & \Delta\mathbf{R}_i\Delta\mathbf{v}_j + \Delta\mathbf{v}_i & \Delta\mathbf{R}_i\Delta\mathbf{p}_j + \Delta\mathbf{v}_i\Delta t_j + \Delta\mathbf{p}_i \\ 0 & 1 & \Delta t_i + \Delta t_j \\ 0 & 0 & 1 \end{bmatrix} \tag{4.24}$$

i els elements invers i neutre també:

$$\begin{aligned}\Delta^{-1} &= \begin{bmatrix} \Delta\mathbf{R}^\top & -\Delta\mathbf{R}^\top\Delta\mathbf{v} & -\Delta\mathbf{R}^\top(\Delta\mathbf{p}-\Delta\mathbf{v}\Delta t) \\ 0 & 1 & -\Delta t \\ 0 & 0 & 1 \end{bmatrix} \\ \Delta_\varepsilon &= \begin{bmatrix} \mathcal{I} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\end{aligned} \tag{4.25}$$

Aquestes deltes viuen a l'espai $\mathbb{R}^2 \times \mathbb{R}^2 \times SO(2) \times \mathbb{R}$, que és una varietat, i per tant tenim un grup de Lie per les deltes de la IMU.

4.4.2 L'espai tangent

Per caracteritzar l'espai tangent procedirem com hem explicat al capítol de teoria de Lie (capítol 3). Definint les variacions instantànies $\mathfrak{v} =: \frac{\partial\Delta\mathbf{p}}{\partial\xi}$, $\mathfrak{a} := \frac{\partial\Delta\mathbf{v}}{\partial\xi}$, $\mathfrak{s} := \frac{\partial\Delta t}{\partial\xi}$, i la matriu $[\omega] := \Delta\mathbf{R}^\top\dot{\Delta\mathbf{R}}$, trobem que l'element $\boldsymbol{\tau}^\wedge$ és

$$\boldsymbol{\tau}^\wedge = \Delta^{-1}\dot{\Delta} = \begin{bmatrix} [\omega] & \Delta\mathbf{R}^\top\mathfrak{a} & \Delta\mathbf{R}^\top(\mathfrak{v}-\mathfrak{s}\Delta\mathbf{v}) \\ 0 & 0 & \mathfrak{s} \\ 0 & 0 & 0 \end{bmatrix} \tag{4.26}$$

i que l'única restricció que tenim és que $[\omega]$ sigui antisimètrica. Denotarem això per $[\omega]_\times$ a través de l'operador antisimètric.

L'equació (4.26) ens dóna l'estructura de $\boldsymbol{\tau}^\wedge = \Delta^{-1}\dot{\Delta}$. Per trobar l'estructura de l'àlgebra de Lie, mirem com és aquest element a la identitat:

$$\Delta_\varepsilon^{-1}\dot{\Delta} = \begin{bmatrix} [\omega]_\times & \mathfrak{a} & \mathfrak{v} \\ 0 & 0 & \mathfrak{s} \\ 0 & 0 & 0 \end{bmatrix} = \boldsymbol{\tau}^\wedge \tag{4.27}$$

Per més detall sobre aquest desenvolupament, veure l'apèndix. B.2.

Tenim la següent EDO $\forall\Delta$:

$$\boldsymbol{\tau}^\wedge = \Delta^{-1}\dot{\Delta} \Longrightarrow \dot{\Delta} = \Delta\boldsymbol{\tau}^\wedge \tag{4.28}$$

i denotarem a partir d'ara $\boldsymbol{\tau} = [\mathfrak{v}, \mathfrak{a}, \omega, \mathfrak{s}]^\top \in \mathbb{R}^6$ com el vector associat a l'àlgebra de Lie $\boldsymbol{\tau}^\wedge$, seguint la notació de (4.23).

4.4.3 L'exponencial

Solucionant l'EDO (4.28), on suposem que $\boldsymbol{\tau}^\wedge$ és constant, arribem a l'aplicació exponencial:

$$\Delta(\xi) = \Delta(0)\exp\left(\begin{bmatrix} [\omega]_\times & \mathfrak{a} & \mathfrak{v} \\ 0 & 0 & \mathfrak{s} \\ 0 & 0 & 0 \end{bmatrix}\cdot\xi\right) \tag{4.29}$$

on $\Delta(0) = I$.

Per trobar la forma tancada considerem l'exponencial matricial en $\xi = \Delta t$: $\Delta(\Delta t) = \exp(A) = \sum_{k\geq 0} \frac{1}{k!} A^k$, on $A = \boldsymbol{\tau}^\wedge \Delta t$.

Definint per conveniència les magnituds $[\theta]_\times := [\omega]_\times \Delta t$, $\nu := \mathfrak{a}\Delta t$, $\rho := \mathfrak{v}\Delta t$, trobem per inducció la següent expressió per les potències d'A:

$$A := \dot{\Delta}\Delta t = \begin{bmatrix} [\theta]_\times & \nu & \rho \\ 0 & 0 & \mathfrak{s}\Delta t \\ 0 & 0 & 0 \end{bmatrix}, \qquad A^0 = \begin{bmatrix} \mathcal{I} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$A^2 = \begin{bmatrix} [\theta]_\times^2 & [\theta]_\times \nu & [\theta]_\times \rho + \nu\mathfrak{s}\Delta t \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, A^3 = \begin{bmatrix} [\theta]_\times^3 & [\theta]_\times^2 \nu & [\theta]_\times^2 \rho + [\theta]_\times \nu\mathfrak{s}\Delta t \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\vdots$$

$$A^k = \begin{bmatrix} [\theta]_\times^k & [\theta]_\times^{k-1} \nu & [\theta]_\times^{k-1} \rho + [\theta]_\times^{k-2} \nu\mathfrak{s}\Delta t \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.30}$$

Definint ara $\mathcal{R} := \sum_{k\geq 0} \frac{1}{k!} [\theta]_\times^k = \exp([\theta]_\times)$, $\mathcal{Q} := \sum_{k\geq 1} \frac{1}{k!} [\theta]_\times^{k-1}$, $\mathcal{P} := \sum_{k\geq 2} \frac{1}{k!} [\theta]_\times^{k-2}$ ens queda:

$$\Delta(\Delta t) = \exp(A) = \begin{bmatrix} \mathcal{R} & \mathcal{Q}\nu & \mathcal{Q}\rho + \mathfrak{s}\mathcal{P}\nu\Delta t \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathcal{R} & \mathcal{Q}\mathfrak{a}\Delta t & \mathcal{Q}\mathfrak{v}\Delta t + \mathfrak{s}\mathcal{P}\mathfrak{a}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \tag{4.31}$$

Aquesta expressió encara depèn de $\mathfrak{v}$, que no és part de les mesures de la IMU. Ens fixem, però, en uns quants detalls:

- Per integrar l'EDO (4.28) hem suposat que $(\mathfrak{a}, \mathfrak{v}, \omega)$ eren constants durant el pas d'integració.
- A la definició de les deltes, havíem dit que eren la diferència en moviment respecte la referència en caiguda lliure (sense acceleració pel nostre cas 2D). A l'inici del període d'integració, la diferència de velocitat entre la referència i la IMU és 0, és a dir, $\mathfrak{v} = 0$.

A més a més, $\mathfrak{a}$ i ω corresponen a les dades que ens dóna la IMU. Havíem suposat a 4.3.3 que aquestes magnituds eren constants durant el període de mostreig de la IMU. Per tant, podem considerar $\mathfrak{v} = 0$ durant tot el pas d'integració, si considerem que aquest pas és més curt o igual que el temps de mostreig de la IMU.

Considerem també el significat de $\mathfrak{s} = \frac{\partial \Delta t}{\partial \xi}$. $\mathfrak{s}$ ens diu com de ràpid evoluciona Δt. És a dir, si prenem $\mathfrak{s} = 1$ estarem dient que les Δt canvien al mateix ritme que el nostre temps t[3]. L'expressió per la delta és en aquest cas:

$$\Delta = \begin{bmatrix} \mathcal{R} & \mathcal{Q}\mathfrak{a}\Delta t & \mathcal{P}\mathfrak{a}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \tag{4.32}$$

amb l'àlgebra de Lie

$$\boldsymbol{\tau}^{\wedge} = \begin{bmatrix} [\omega]_{\times} & \mathfrak{a} & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.33}$$

Les expressions tancades de $\mathcal{R}$, $\mathcal{Q}$ i $\mathcal{P}$ són les següents:

$$\mathcal{R} = \cos(\theta) I + \sin(\theta) [1]_{\times} = \mathbf{R}(\theta) \tag{4.34}$$

$$\mathcal{Q} = \frac{\sin(\theta)}{\theta} I + \frac{1 - \cos(\theta)}{\theta} [1]_{\times} \tag{4.35}$$

$$\mathcal{P} = \frac{1 - \cos(\theta)}{\theta^2} I + \frac{\theta - \sin(\theta)}{\theta^2} [1]_{\times} \tag{4.36}$$

on $\mathbf{R}(\theta)$ és la matriu de rotació d'angle θ. Per un desenvolupament detallat d'aquestes expressions, veure l'apèndix. B.3.

Observem que per valors de θ prou petits $\mathcal{Q} \approx I$, $\mathcal{P} \approx \frac{1}{2} I$, i si identifiquem $\theta = \omega \Delta t$ (és a dir, amb una velocitat angular i un pas de temps prou petits):

$$\Delta = \begin{bmatrix} \exp\left([\omega]_{\times} \Delta t\right) & \mathfrak{a}\Delta t & \frac{1}{2}\mathfrak{a}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \tag{4.37}$$

Que coincideix amb la delta que havíem definit a partir de les magnituds inercials quan $\Delta t = \delta t$ (4.18), i amb la que surt fent el grup de Lie per blocs. Si no fem aquesta aproximació d'angle petit, tenim una definició per la delta actual més acurada, ja que té en compte interaccions entre les parts de la delta.

4.4.4 Preintegració incremental en el grup de Lie

Com hem vist a la secció 4.3.3, la preintegració incremental de les deltes ve donada per la composició de grup. En termes de l'exponencial que hem trobat a l'apartat anterior:

$$\Delta_{ik} = \Delta_{ij} \delta_{jk} = \Delta_{ij} \exp(\boldsymbol{\tau}_{jk}^{\wedge} \delta t) = \Delta_{jk} \mathrm{Exp}(\boldsymbol{\tau}_{jk} \delta t) \tag{4.38}$$

[3] Valors de $\mathfrak{s} \neq 1$ podrien ser usats per usats en sistemes que utilitzen un rellotge que va a un ritme diferent del temps real.

On $\boldsymbol{\tau}_{jk} = [\mathbf{0}, \mathfrak{a}_k, \omega_k, 1]$ és de l'espai tangent a Δ_{ij} expressat en referència local. Aquesta preintegració depèn només de les dades que ens dona la nostra IMU $(\mathfrak{a}_k, \omega_k)$ i del temps de mostreig δt, tal com volíem.

4.4.5 Jacobianes

A causa del que hem mencionat al principi d'aquesta secció, desenvoluparem les jacobianes tant pel grup de Lie compacte com pel grup de Lie per blocs. Es poden fer servir indistintament, però s'ha d'anar amb compte de no barrejar-les. És a dir, quan apliquem la regla de la cadena i fem servir les jacobianes en la propagació de covariància i correcció de les deltes, haurem d'escollir quina definició fer servir.

Les jacobianes trivials són les corresponents a les magnituds inercials (4.11):

Compacta	**Blocs**
$\mathbf{J}^{\mathbf{b}}_{\mathbf{b}_m} = I_3$ (4.39)	$\mathbf{J}^{\mathbf{b}}_{\mathbf{b}_m} = I_3$ (4.42)
$\mathbf{J}^{\mathbf{b}}_{\mathbf{b}_b} = -I_3$ (4.40)	$\mathbf{J}^{\mathbf{b}}_{\mathbf{b}_b} = -I_3$ (4.43)
$\mathbf{J}^{\mathbf{b}}_{\mathbf{b}_n} = -I_3$ (4.41)	$\mathbf{J}^{\mathbf{b}}_{\mathbf{b}_n} = -I_3$ (4.44)

on hem fet servir la notació $\mathbf{b}_m = (\mathbf{a}_m, \omega_m)$, $\mathbf{b}_b = (\mathbf{a}_b, \omega_b)$ i $\mathbf{b}_n = (\mathbf{a}_n, \omega_n)$. És normal que donin igual ja que no són jaocbianes de les deltes.

Sigui $\boldsymbol{\tau} = [\mathfrak{v}, \mathfrak{a}, \omega, \mathfrak{s}]^\top$ el vector tangent a la delta del pas actual $\boldsymbol{\delta}$. Si no tenim una expressió tancada de la seva Exponencial, $[\delta\mathbf{p}, \delta\mathbf{v}, \delta\mathbf{R}, \delta t]^\top =: \boldsymbol{\delta} = \mathrm{Exp}(\boldsymbol{\tau}\delta t)$, la jacobiana de $\boldsymbol{\delta}$ respecte $\boldsymbol{\tau}$ és, seguint la notació del capítol 3:

Compacta	**Blocs**
$\mathbf{J}^{\boldsymbol{\delta}}_{\boldsymbol{\tau}} := \dfrac{D\boldsymbol{\delta}}{D\boldsymbol{\tau}} = \dfrac{D\,\mathrm{Exp}(\boldsymbol{\tau}\delta t)}{D\boldsymbol{\tau}} = \mathbf{J}_r(\boldsymbol{\tau}\delta t)\delta t$ (4.45)	$\mathbf{J}^{\boldsymbol{\delta}}_{\boldsymbol{\tau}} = \begin{bmatrix} \mathcal{Q}\delta t & \mathfrak{s}\mathcal{P}\delta t^2 & \mathcal{Q}'\,[1]_\times\,\mathfrak{v}\delta t^2 + \mathcal{P}'\,[1]_\times\,\mathfrak{a}\delta t^3 & 0 \\ 0 & \mathcal{Q}\delta t & \mathcal{Q}'\,[1]_\times\,\mathfrak{a}\delta t^2 & 0 \\ 0 & 0 & -\mathcal{R}\delta t^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ (4.46)

on $\mathbf{J}_r$ és la Jacobiana per la dreta de la varietat. D'aquesta jacobiana no hem trobat, per ara, una forma tancada. Això impedeix que fem servir les jacobianes de la versió compacta en la posterior implementació. En la jacobiana per blocs, $\mathcal{R}$, $\mathcal{Q}$, $\mathcal{P}$, són les funcions de (4.31); $\mathcal{R}'$, $\mathcal{Q}'$, $\mathcal{P}'$ les seves derivades respecte θ; totes elles avaluades en $\omega\delta t$.

En el nostre cas, però, sí tenim una expressió tancada quan $\mathfrak{v} = 0$ i $\mathfrak{s} = 1$ corresponent a considerar el vector tangent que ve de les magnituds inercials. Per fer això, prenem la definició d'exponencial de (4.32) amb només dues variables, les de les magnituds inercials:

Compacta	Blocs
$\mathbf{J}^{\boldsymbol{\delta}}_{\mathbf{b}} = \mathbf{J}^{\boldsymbol{\delta}}_{\boldsymbol{\tau}}\mathbf{J}^{\boldsymbol{\tau}}_{\mathbf{b}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{J}_r(\boldsymbol{\tau}\delta t)\delta t$ (4.47)	$\mathbf{J}^{\boldsymbol{\delta}}_{\mathbf{b}} = \begin{bmatrix} \mathcal{P}\delta t^2 & \mathcal{P}'\left[1\right]_\times \mathbf{a}\delta t^2 \\ \mathcal{Q}\delta t & \mathcal{Q}'\left[1\right]_\times \mathbf{a}\delta t \\ 0 & -\mathcal{R}\delta t \\ 0 & 0 \end{bmatrix}$ (4.48)

on $\mathcal{R}$, $\mathcal{Q}$, $\mathcal{P}$, són una altra vegada les funcions de (4.31); $\mathcal{R}'$, $\mathcal{Q}'$, $\mathcal{P}'$ les seves derivades respecte θ; totes elles avaluades en $\omega\delta t$. Aquesta vegada, però, tenim expressions tancades per totes elles, que podem trobar a l'apèndix. B.3.

Per la composició $\Delta_{ik} = \Delta_{ij}\delta_{jk}$, tenim les següents jacobianes:

Compacta	Blocs
$\mathbf{J}^{\Delta_{ik}}_{\Delta_{ij}} = \mathbf{J}^{\Delta_{ij}\delta_{jk}}_{\Delta_{ij}} = \mathbf{Ad}_{\boldsymbol{\delta}_{jk}}{}^{-1}$ (4.49) $\mathbf{J}^{\Delta_{ik}}_{\delta_{jk}} = \mathbf{J}^{\Delta_{ij}\delta_{jk}}_{\delta_{jk}} = I_6$ (4.50)	$\mathbf{J}^{\Delta_{ik}}_{\Delta_{ij}} = \begin{bmatrix} I_2 & I_2\delta t_{jk} & \delta\mathbf{p}_{jk} & 0 \\ 0 & I_2 & \delta\mathbf{v}_{jk} & 0 \\ 0 & 0 & \delta\mathbf{R}_{jk} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ (4.51) $\mathbf{J}^{\Delta_{ik}}_{\delta_{jk}} = \begin{bmatrix} \Delta\mathbf{R}_{ij} & 0 & 0 & \Delta\mathbf{v}_{ij} \\ 0 & \Delta\mathbf{R}_{ij} & 0 & 0 \\ 0 & 0 & \Delta\mathbf{R}_{ij} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ (4.52)

4.4.6 Propagació incremental de la covariància

Denotant per $\boldsymbol{\Sigma}_{\boldsymbol{\delta}_{jk}}$ la covariància de la delta actual, $\boldsymbol{\Sigma}_{\Delta_{ij}}$ la de la delta preintegrada, i $\boldsymbol{\Sigma}_n$ la del soroll del mesurament, les covariàncies es propaguen de la següent manera:

$$\boldsymbol{\Sigma}_{\Delta_{ik}} = \mathbf{J}^{\Delta_{ik}}_{\Delta_{ij}}\boldsymbol{\Sigma}_{\Delta_{ij}}\mathbf{J}^{\Delta_{ik}}_{\Delta_{ij}}{}^{\top} + \mathbf{J}^{\Delta_{ik}}_{\delta_{kj}}\boldsymbol{\Sigma}_{\boldsymbol{\delta}_{jk}}\mathbf{J}^{\Delta_{ik}}_{\delta_{kj}}{}^{\top} \tag{4.53}$$

on $\boldsymbol{\Sigma}_{\boldsymbol{\delta}_{jk}} = \mathbf{J}^{\boldsymbol{\delta}_{jk}}_{\mathbf{b}_n}\boldsymbol{\Sigma}_n\mathbf{J}^{\boldsymbol{\delta}_{jk}}_{\mathbf{b}_n}{}^{\top}$, i obtenim $\mathbf{J}^{\boldsymbol{\delta}_{jk}}_{\mathbf{b}_n} = \mathbf{J}^{\boldsymbol{\delta}_{jk}}_{\mathbf{b}}\mathbf{J}^{\mathbf{b}}_{\mathbf{b}_n}$ amb la regla de la cadena.

4.4.7 La jacobiana de delta respecte el biaix, correcció de les deltes

Siguin $\bar{\Delta}$ i $\bar{\mathbf{b}}_b$ la delta preintegrada i el biaix utilitzat durant aquesta preintegració. Actualitzarem la delta a cada iteració de l'optimització utilitzant la nova estimació del biaix $\mathbf{c}$, mitjançant la linealització

$$\Delta = \bar{\Delta} \oplus \mathbf{J}^{\Delta}_{\mathbf{b}_b}(\mathbf{b}_b - \bar{\mathbf{b}}_b) \tag{4.54}$$

Per trobar una manera de construir aquesta jacobiana incrementalment sigui $\Delta = \Delta_{ik} = \Delta_{ij}\delta_{jk}$, i apliquem la regla de la cadena:

$$\mathbf{J}^{\Delta_{ik}}_{\mathbf{b}_b} = \mathbf{J}^{\Delta_{ij}\delta_{jk}}_{\mathbf{b}_b} = \mathbf{J}^{\Delta_{ij}\delta_{jk}}_{\Delta_{ij}}\mathbf{J}^{\Delta_{ij}}_{\mathbf{b}_b} + \mathbf{J}^{\Delta_{ij}\delta_{jk}}_{\delta_{jk}}\mathbf{J}^{\delta_{jk}}_{\mathbf{b}_b} \tag{4.55}$$

amb $\mathbf{J}^{\delta_{jk}}_{\mathbf{b}_b} = \mathbf{J}^{\delta_{jk}}_{\mathbf{b}}\mathbf{J}^{\mathbf{b}}_{\mathbf{b}_b}$ i $\mathbf{J}^{\Delta_{ij}}_{\mathbf{b}_b}$ la jacobiana de l'anterior pas incremental, que comença per $\mathbf{J}^{\delta_{ii}}_{\mathbf{b}_b} = \mathbf{0}$.

4.5 Pipeline

Finalment, resumim a continuació com utilitzem els objectes definits al llarg d'aquest capítol:

1. La IMU fa una mesura. Aquesta dada és corregida utilitzant el biaix estimat, donant les magnituds inercials: $f(\mathbf{z}_k, \bar{\mathbf{b}_b}) = \mathbf{b}_{jk}$ (4.11).
2. Es calcula la delta actual a partir de les magnituds inercials: $\delta_{jk} = \mathrm{Exp}(\boldsymbol{\tau}_{jk}\delta t)$ (4.38).
3. Es preintegra la delta $\bar{\Delta}_{ik} = \bar{\Delta}_{ij}\delta_{jk}$ (4.38), es propaga la covariància $\boldsymbol{\Sigma}_{\Delta_{ij}}$ (4.53) i la jacobiana del biaix $\mathbf{J}^{\Delta_{ik}}_{\mathbf{b}_b}$ (4.55).
4. Es repeteixen els passos 1-3 fins que es genera un keyframe $\mathbf{x}_p$. Es crea el factor que el relaciona amb l'anterior keyframe $\mathbf{x}_i$, que conté el resultat d'aquests càlculs incrementals: $\bar{\Delta}_{ip}$, $\boldsymbol{\Sigma}_{\Delta_{ip}}$, $\mathbf{J}^{\Delta_{ip}}_{\mathbf{b}_b}$.
5. A cada iteració de l'optimització corregim la delta de cada factor afectat usant la nova estimació del biaix (4.54) i tornem a computar l'error del factor amb el nou biaix. $\mathbf{e}_{ip} = \bar{\Delta}_{ip}(\bar{\mathbf{c}}_i) \ominus (\mathbf{x}_p \diamondminus \mathbf{x}_i)$ (4.4).

5 WOLF

WOLF (**W**ind**O**wed **L**ocalization **F**rames) [Solà* et al., 2021] és una llibreria desenvolupada a l'IRI per l'estimació d'estat de robots, basada en grafs de factors. Està dissenyada modularment de manera que nous algorismes i sensors poden ser afegits de manera escalable. Tenim la llibreria central anomenada *WOLF-core* que conté les capacitats bàsiques, aquesta és estesa a través dels plugins, que són llibreries dependents de WOLF-core amb implementacions per nous sensors/algorismes. La feina detallada en aquest treball ha sigut implementada a WOLF a través d'un plugin, desenvolupant la capacitat de resoldre problemes 2D.

5.1 Estructura, l'arbre de WOLF

WOLF organitza les diferents parts que apareixen en problemes de navegació:

1. Hardware i Software:
    - Sensors
    - Processadors de dades
2. Trajectòria (s'actualitza segons avança el temps):
    - Frames
    - Captures de sensors
    - Features o mesures extretes de les captures
    - Factors enllaçant les features a altres entitats. N'hi ha de diversos tipus:
        - de moviment i de loop closure: enllacen a altres Frames
        - de landmark: enllacen a landmarks del mapa
        - de calibratge: enllacen a paràmetres dels sensors
        - absoluts: enllacen només una entitat
        - combinacions dels anteriors
3. Mapa: Landmarks

Aquesta estructura està representada en l'arbre de WOLF (Fig. 5.1), que és on es guarda tota la informació del problema. Consta de les classes base que reprodueixen tots els elements

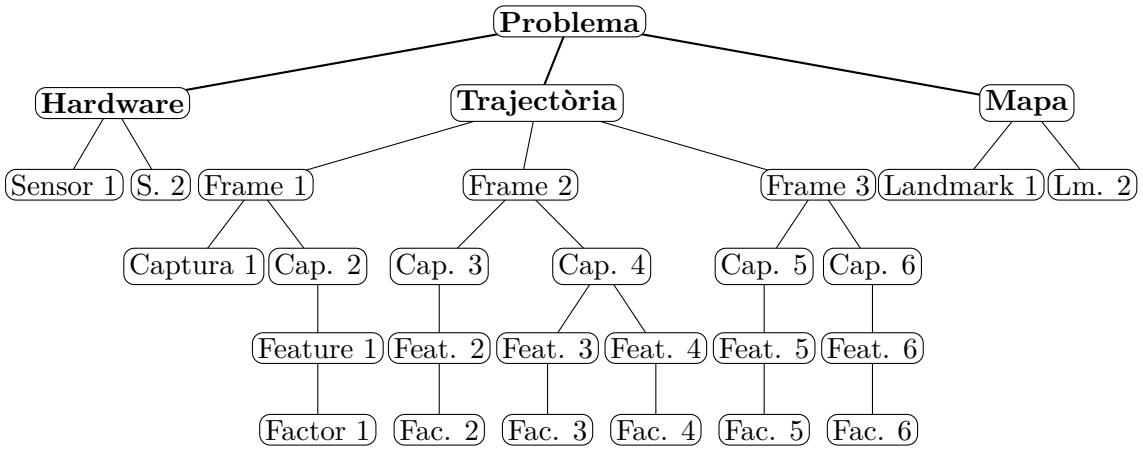

FIGURA 5.1: Arbre de WOLF bàsic on apareixen els membres de l'estructura

del problema robòtic. El node arrel és el **Problema**, d'on surten els nodes **Hardware**, **Trajectòria** i **Mapa**, les tres components principals del nostre problema:

- **Hardware:** És una llista de sensors, on cada sensor té:
  - Un o més blocs d'estat pels paràmetres **extrínsecs**: posició i orientació en el robot.
  - Un o més blocs d'estat pels paràmetres **intrínsecs**: típicament paràmetres de calibratge.
  - Una llista de processadors als quals està relacionat el sensor.

  Els blocs d'estat són una classe que conté petits vectors d'estat que representen entitats físiques que volem estimar.
- **Trajectòria:** és una llista de referències a *Frames*, cadascun descrivint l'estat del robot en un temps diferent.
- **Mapa:** és una llista de referències a *Landmarks*, cadascun descrivint l'estat d'algun objecte de l'entorn.

. La majoria dels nodes que segueixen són contenidors de dades:

- Els nodes de sensor contenen els paràmetres extrínsecs i intrínsecs.
- Els Frames formen l'estat i contenen captures.
  - Les Captures contenen mesures dels sensors, Features (mesures mètriques derivades de les dades), així com els Factors associats.
- els Landmarks contenen l'estat/descripció de l'objecte de l'entorn.

Gairebé tot el processament es duu a terme en 3 nodes: Processadors (front-end), Factors (back-end) i Problema (gestió de l'arbre i comunicació amb el solver). Les dades dels sensors són processades en el front-end pels processadors. El front-end és també l'encarregat de generar l'arbre de WOLF, a partir del qual tindrem el graf del problema, poblant-lo de nous

nodes (Frames, Captures, Landmarks,etc) segons sigui necessari. Aquest graf després passa pel back-end, on el solver fa l'optimització descrita al capítol 2.

Cadascun dels nodes de les branques de l'arbre pot tenir un nombre arbitrari de fills, i aquests tenen accés al seu node pare. Hi ha, però, certes connexions entre nodes que trenquen aquesta estructura d'arbre i ens permeten crear una graf de factors que representa al nostre problema, com podem veure a la Fig. 5.2. Aquestes connexions són les següents:

- Les Captures tenen accés al sensor que les ha creat
- Els Factors tenen accés als nodes que continguin els blocs d'estat necessaris per computar el seu residu.
- Els nodes tenen accés a qualsevol factor que involucri un dels seus blocs d'estat

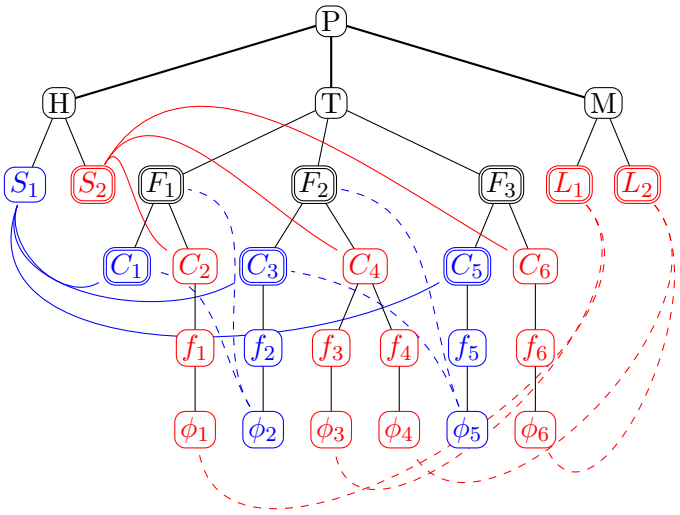

FIGURA 5.2: Arbre de WOLF amb les relacions que el fan equivalent a un graf de factors. En vermell mesures visuals de Landmarks, en blau mesures de la IMU. Tota Captura té un punter cap al sensor que l'ha creat (sòlida), i els Factors ϕ tenen apuntadors (discontinua) cap als nodes que apareixen al seu model de mesura. En doble línia estan indicats els nodes que contenen un bloc d'estat, paràmetres extrínsecs en el cas de la càmera S_2 i biaixos en les captures de la IMU C_1, C_2, C_3. Veure Fig. 5.4 pel graf de factors equivalent

.

5.1.1 Inicialització del problema

Un problema de SLAM comença amb el robot preparat per a dur a terme la seva missió. Al nostre arbre, això correspon a tenir tota la branca de Hardware inicialitzada, i un Frame dins la Trajectòria amb les condicions inicials del robot. Mitjançant WOLF-autoconf podem utilitzar arxius YAML per tal de descriure el hardware i aquesta situació inicial. La Fig. 5.3 descriu aquest procés. Els arxius YAML contenen els sensors i els seus paràmetres, els processadors i els seus paràmetres, el solver i els seus paràmetres, el gestor de l'arbre i els seus paràmetres, i l'estat inicial del robot. Aquests arxius també li indiquen a WOLF on trobar les diferents peces. WOLF carrega durant el runtime tots els plugins i registra les classes derivades en factories específiques a core. Això vol dir que podem preparar diferents

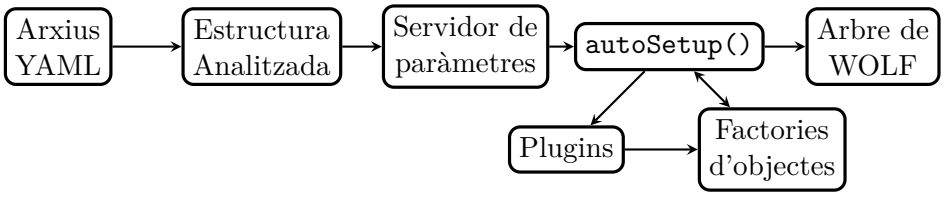

FIGURA 5.3: Pipeline de configuració mitjançant arxius YAML. Els arxius YAML són analitzats i la seva estructura guardada en un servidor de paràmetres. El mètode *autoSetup()* llegeix d'aquest servidor i crida als plugins necessaris, els creadors d'objectes dels quals són registrats a les factories de core. Finalment utilitza tot això per inicialitzar l'arbre de WOLF.

problemes d'estimació sense haver d'escriure codi ni tornar a compilar, simplement cambiant aquests arxius YAML.

5.1.2 Processadors

La resta de l'arbre la construeixen incrementalment els processadors segons reben dades noves. Són responsables de:

- Extreure Features de les dades contingudes en Captures.
- Associar Features amb altres nodes de l'arbre.
- Decidir sobre la creació de Frames:
  - Crear Factors per cada Feature associada.
  - Crear nous Landmarks al mapa.
  - Avisar de la creació del Frame.
- Unir-se a Frames generats per altres processadors.
- Tancar loops.

Els processadors creen nous nodes, però no tenen la capacitat d'eliminar-los. Per tant, cal un gestor que pugui eliminar nodes segons sigui necessari per mantenir l'arbre d'una extensió tractable. A WOLF-core hi ve inclòs un gestor que fa servir finestres lliscants sobre els Frames, però és possible implementar-ne d'altres amb criteris diferents.

Hi ha molta varietat en algorismes de processament, però en SLAM podem identificar-ne tres tipus principals: seguiment del moviment, seguiment de features, i tancament de loops. El processador programat durant aquest treball és del primer tipus, i en farem un petit resum, tot i que ja l'hem descrit en detall als altres capítols.

Preintegradors de moviment amb autocalibratge

Basant-nos en la teoria general de preintegració i afegint-hi les deltes i jacobianes de la teoria de Lie, tenim la següent pipeline implementada genèricament a WOLF-core:

1. Precalibratge de les dades de moviment: $\mathbf{v} = f(\mathbf{u}, \bar{\mathbf{c}})$, $\mathbf{J}^{\mathbf{v}}_{\mathbf{u}}$, $\mathbf{J}^{\mathbf{v}}_{\bar{\mathbf{c}}}$

2. Computació de la delta actual: $\delta = g(\mathbf{v})$, $\mathbf{J}^{\delta}_{\mathbf{v}}$
3. Preintegració de la delta: $\bar{\Delta} \longleftarrow \bar{\Delta} \circ \delta$, $\mathbf{J}^{\Delta}_{\Delta}$, $\mathbf{J}^{\Delta}_{\delta}$
4. Integració de la covariància: $\mathbf{Q}_{\Delta} \longleftarrow \mathbf{J}^{\Delta}_{\Delta}\mathbf{Q}_{\Delta}\mathbf{J}^{\Delta}_{\Delta}{}^{\top} + \mathbf{J}^{\Delta}_{\delta}\mathbf{J}^{\delta}_{\mathbf{v}}\mathbf{J}^{\mathbf{v}}_{\mathbf{u}}\mathbf{Q}_{\mathbf{u}}(\mathbf{J}^{\Delta}_{\delta}\mathbf{J}^{\delta}_{\mathbf{v}}\mathbf{J}^{\mathbf{v}}_{\mathbf{u}})^{\top}$
5. Integració de la jacobiana respecte dels paràmetres de calibratge: $\mathbf{J}^{\Delta}_{\bar{\mathbf{c}}} \longleftarrow \mathbf{J}^{\Delta}_{\Delta}\mathbf{J}^{\Delta}_{\bar{\mathbf{c}}} + \mathbf{J}^{\Delta}_{\delta}\mathbf{J}^{\delta}_{\mathbf{v}}\mathbf{J}^{\mathbf{v}}_{\bar{\mathbf{c}}}$

On $\mathbf{u}$ són les dades de moviment del sensor, $\bar{\mathbf{c}}$ l'estimació inicial dels paràmetres de calibratge, $\mathbf{v}$ les dades de moviment calibrades, δ la delta de moviment actual, $\bar{\Delta}$ la delta de moviment preintegrada, $\mathbf{J}^{y}_{x}$ les jacobianes computades segons la teoria de Lie, i $\mathbf{Q}_x$ la covariància de x.

Es publiquen estats a la mateixa freqüència d'entrada de les dades, amb $\mathbf{x}_t = \mathbf{x}_i \boxplus \bar{\Delta}_{it}$ on $\mathbf{x}_i$ és l'últim Frame. Les classes derivades d'aquest tipus de processador només han d'implementar la funció $g(f(\mathbf{u}, \bar{\mathbf{c}}))$, la composició $\circ$ i l'operador $\boxplus$ per tal d'implementar un nou processador d'aquest tipus.

5.1.3 Factors

Pel que fa als Factors, la delta preintegrada $\bar{\Delta}$ pot ser corregida per valors dels paràmetres de calibratge $\mathbf{c}$ diferents als inicials $\bar{\mathbf{c}}$ mitjançant la linealització $\Delta(\mathbf{c}) = \bar{\Delta} \oplus \mathbf{J}^{\Delta}_{\mathbf{c}}(\mathbf{c} - \bar{\mathbf{c}})$. El residu entre dos Frames consecutius $\mathbf{r}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{c}) = \mathbf{Q}^{\mathrm{T}/2}_{\Delta}(\Delta(c) \ominus (\mathbf{x}_j \boxminus \mathbf{x}_i))$ fa que puguem observar el paràmetre $\mathbf{c}$. Això ens permet, per exemple, actualitzar l'estimació dels biaixos de la nostra IMU, però notem que en el cas general és vàlid per qualsevol paràmetre de qualsevol sensor.

5.1.4 Graf i Solver

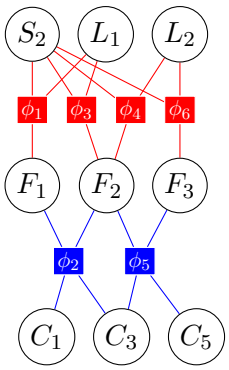

FIGURA 5.4: El graf de factors equivalent a la Fig. 5.2. Aquest graf ens permet estimar els paràmetres extrínsecs de la càmera pels blocs d'estat a S2 i el biaix de la IMU a les Captures C1, C3 i C5.

L'arbre de WOLF és equivalent a un graf de factors. Podem veure un exemple de graf en la 5.4, Els nodes d'estat d'aquest estan continguts en els nodes Sensor, Frame, Captura i Landmark. WOLF permet designar certs nodes d'estat com a fixos per excloure'ls de l'optimització. Els nodes de factor del graf vénen dels nodes Factor de l'arbre. Aquests s'encarreguen de calcular un residu i apuntar cap a tots els blocs d'estat que apareixen al

seu model de mesura. Opcionalment, també poden funcions de pèrdua per permetre una estimació robusta davant outliers. Com el solver requereix de les jacobianes, WOLF inclou Factors base que poden calcular jacobianes numèriques en cas de no tenir-ne d'analítiques.

Finalment el solver no lineal efectua les següents passes a cada iteració:

1. Avalua els residus de tots els Factors i computa les jacobianes respecte els blocs d'estat.
2. Ajunta totes les jacobianes en una matriu jacobiana gran $\mathbf{J}$ i tots els residus en un vector $\mathbf{r}$.
3. Obté el pas d'optimització $\Delta\mathbf{x}$.
4. Actualitza l'estat $\mathbf{x} \Longleftarrow \mathbf{x} \oplus \Delta\mathbf{x}$ amb un operador $\oplus$ adient
5. Itera fins a convergir numèricament.

WOLF pot ser utilitzat amb qualsevol solver iteratiu no lineal per grafs, com Google Ceres, GTSAM, G2O, SLAM++, etc. La compatibilitat es dóna a través de wrappers específics per cada solver, WOLF core ve amb un wrapper preparat per Google Ceres.

5.2 ROS

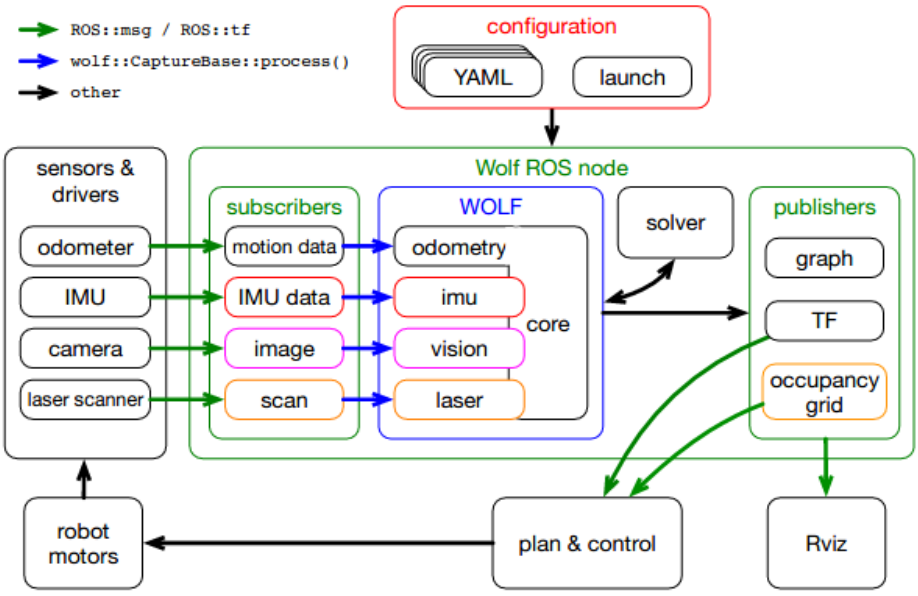

FIGURA 5.5: Pipeline de WOLF usant ROS d'intermediari. Cada tipus de sensor utilitza un plugin (odometry forma part de core) i el subscriber del mateix color. Els publishers creen missatges de ROS amb dades pel control i la visualització.

ROS [Quigley et al.] (Robot Operating System) és un framework que permet la correcta comunicació entre el hardware i els programes d'un robot. Es basa també en una estructura de graf, on els programes són els nodes. Els nodes es poden comunicar amb els altres mitjançant un sistema de missatges. Els nodes publiquen *topics*, on deixen escrit la informació

que volen compartir, i poden subscriure's a topics d'altres nodes per rebre les dades allà publicades.

A la Fig. 5.5 podem veure una representació bàsica de la pipeline usant ROS com a intermediari entre WOLF i el hardware.

WOLF implementa *subscribers*, que fan servir la subscripció de ROS per rebre dades i convertir-les al format apropiat per les Captures, cridant immediatament el mètode *process()* d'aquestes. Aquest mètode avisa als processadors associats al sensor que hi ha noves dades.

WOLF també implementa uns *publishers*, que s'encarreguen de publicar informació diversa de l'arbre a través de publicacions de ROS. Són principalment missatges relacionats amb la visualització i monitorització, a més de missatges de control si s'han d'efectuar accions al robot.

La peça central de la implementació de WOLF en ROS és el *WOLF ROS node*. Inclou les llistes de subscribers i publishers, el wrapper del solver, i la interacció amb l'arbre de WOLF. Per controlar-lo només cal escriure l'arxiu YAML adient per la nostra configuració, i una *launchfile* bàsica de ROS amb les rutes dels arxius YAML. El node prendrà aquests arxius, carregarà els plugins adients per inicialitzar l'arbre de WOLF, configurarà el solver segons els paràmetres donats, i crearà els publishers i subscribers necessaris.

5.3 El plugin per la IMU

El plugin WOLF-IMU ha estat desenvolupat a l'IRI per estendre les capacitats de WOLF-core per treballar amb sensors del tipus IMU. Hi estava implementat tot el necessari per treballar amb aquests sensors en problemes 3D, però mancava la capacitat de fer-los servir en problemes 2D. La teoria exposada en aquest treball ha estat implementada en aquest plugin per cobrir aquesta necessitat.

La nova classe principal que hem creat és la de *ProcessorImu2D* (`processor_imu2d.h` i `processor_imu2d.cpp`). Conté les funcions que utilitza WOLF per tal de seguir la teoria de preintegració que hem detallat al capítol 4:

- *computeCurrentDelta* genera la delta actual $\boldsymbol{\delta}$ definida a (4.32) a través de l'auxiliar *body2delta()*, i computa la seva covariancia i la seva jacobiana.
- *deltaPlusDelta* implementa la composició del grup de Lie definida a (4.15). Aquesta funció està sobrecarregada amb una versió que a més a més computa les jacobianes.
- *statePlusDelta* implementa l'operador $\oplus$ per tal de poder sumar deltes a estats $\mathbf{x}$.
- *correctDelta* implementa l'actualització de la delta donat un canvi en els biaixos (4.54).

Aquesta classe depèn fortament de la nova llibreria auxiliar `imu2d_tools.h`, on hem implementat les diferents operacions matemàtiques que es fan servir tant en aquest processador com a la resta del plugin:

- *identity()* retorna la delta identitat.
- *inverse()* retorna la delta inversa.
- *compose()* composa dues deltes.
- *between()* donades Δ_1 i Δ_2 retorna una Δ_r tal que $\Delta_2 = \Delta_1 \circ \Delta_r$.
- *composeOverState()* retorna $\mathbf{x}_2 = \mathbf{x}_1 \oplus \Delta$.
- *betweenStates()* donats estats $\mathbf{x}_1$ i $\mathbf{x}_2$ retorna una delta Δ_r tal que $\mathbf{x}_2 = \mathbf{x}_1 \oplus \Delta_r$.
- *log_Imu()* la funció logaritme descrita al capítol 3, porta de la varietat de les deltes a l'espai tangent.
- *exp_Imu()* la funció exponencial descrita al capítol 3, porta de l'espai tangent a la varietat de les deltes.
- *plus()* implementa l'operador $\oplus$ mitjançant `exp_Imu()`.
- *diff()* implementa l'operador $\ominus$ mitjançant `log_Imu()`.
- *body2delta()* construeix una delta actual δ a partir de les magnituds inercials.

A part dels arxius creats específicament pel cas 2D, que hem notat amb el sufix "2d" al nom de l'arxiu, també s'han modificat parts del plugin per tal d'acomodar el cas 2D. A les classes *CaptureImu* i *SensorImu*, per exemple, se'ls ha afegit una comprovació de la dimensió del problema a l'hora de crear el StateBlock, que és de dimensió 3 en el cas 2D però de dimensió 6 en el 3D. Podem trobar una llista dels arxius creats i modificats a l'apèndix. C

5.3.1 Tests Unitaris

A més a més, per tal de poder assegurar la funcionalitat d'aquest codi, hem preparat una sèrie de tests unitaris que comproven que cadascuna de les funcions funcioni com s'espera. Estan localitzats als arxius `gtest_processor_imu2d.cpp` i `gtest_imu2d_tools.cpp`. Els d'aquest segon són molt senzills, ja que les funcions de `imu2d_tools.h` són simples i només comproven que retornin el que han de retornar amb alguns casos límit. D'altra banda, els corresponents al processador simulen un parell de situacions d'estimació de moviment i comparen els resultats amb la solució analítica:

Moviment rectilini uniformement accelerat

La simulació en aquest test és la d'un robot fent un MRUA. S'aconsegueix donant-li al processador dades de IMU corresponents a $\mathbf{a} = (1, 0)$ i velocitat angular $\omega = 0$. Es determina també una posició inicial p_0, una velocitat inicial v_0 i un temps inicial t_0.

Finalment un bucle li passa les dades al processador a intervals de temps determinats, li demana l'estimació de l'estat actual (posició), i la compara amb el que dóna la solució analítica

$$p = p_0 + v_0(t - t_0) + \frac{1}{2}a(t - t_0)^2 \tag{5.1}$$

Si en qualsevol moment la comparació no dona el mateix amb un marge d'error de 10^{-9} el test falla.

Paràbola

En aquest test simulem una trajectòria en forma de paràbola. Com abans, se li dóna al processador unes dades de IMU corresponents a $\mathbf{a} = (1, 2)$ i velocitat angular $\omega = 0$. Es determina també una posició inicial p_0, una velocitat inicial v_0 i un temps inicial t_0.

Finalment, un bucle li passa les dades al processador a intervals de temps determinats, li demana l'estimació de l'estat actual (posició i velocitat), i la compara amb el que dóna la solució analítica

$$p = p_0 + v_0(t - t_0) + \frac{1}{2}a(t - t_0)^2 \tag{5.2}$$

$$v = v_0 + a(t - t_0) \tag{5.3}$$

Si en qualsevol moment la comparació no dóna el mateix amb un marge d'error de 10^{-9}, el test falla.

Moviment Circular

En aquest test simulem una trajectòria en forma de cercle de radi r a velocitat angular constant. Es determina una orientació inicial del robot α, una posició inicial $p_0 = (r, 0)$, una velocitat inicial $v_0 = (0, \omega r)$, i un temps inicial t_0. Li donem al processador unes dades de IMU corresponents a la velocitat angular $\omega = 1$, i l'acceleració $a = (-\cos(\alpha)\omega^2 r, \sin(\alpha)\omega^2 r)$, que correspon a l'acceleració centrípeta que experimentaria la IMU.

Finalment com en els altres tests un bucle li passa les dades al processador a intervals de temps dt determinats, li demana l'estimació de l'estat actual (posició), i la compara amb el que dona la solució analítica

$$p = r(\cos(\omega(t - t_0)), \sin(\omega(t - t_0))) \tag{5.4}$$

Si en qualsevol moment la comparació no dona el mateix amb un marge d'error de 10^{-9} el test falla.

En aquest últim test cal fer notar que si el processador fa servir la definició de delta actual donada per (4.18) el test falla ja que el marge d'error passa a ser d'ordre 10^{-4}. En canvi fent servir la definició trobada a (4.32) sense l'aproximació d'angle petit ens dona prou precisió com per passar el test.

6 Experiments

Sumat als tests unitaris del capítol 5, que validen el funcionament correcte del codi, hem dut a terme una sèrie d'experiments empírics per avaluar el funcionament al mon real. Parlarem primer de la plataforma robòtica utilitzada, i dels altres sistemes que treballen conjuntament amb el plugin de la IMU per estimar l'estat.

6.1 El robot: Ana-Pioneer 3

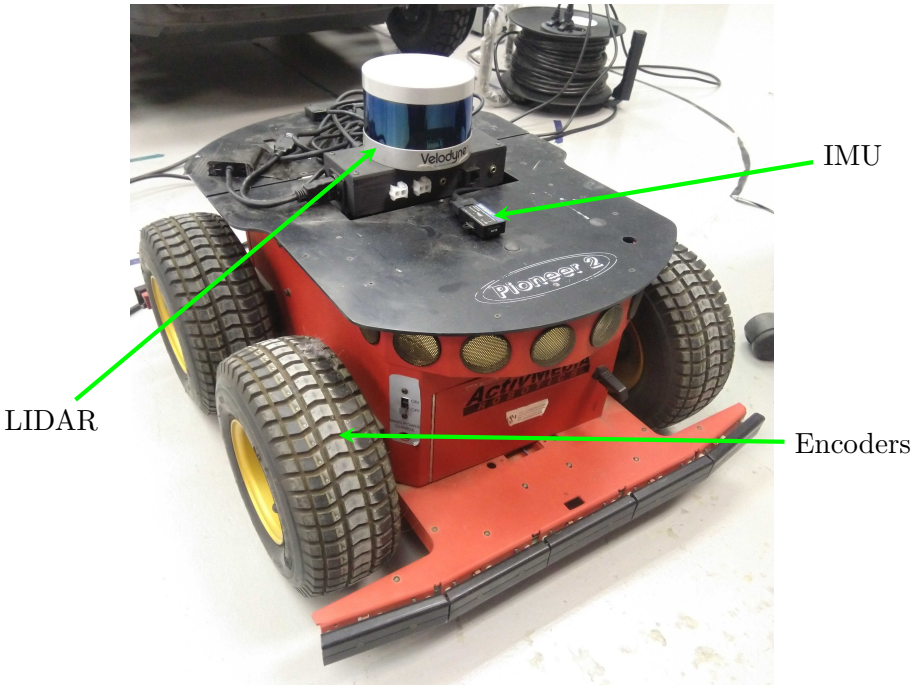

FIGURA 6.1: Els diferents sistemes del robot Ana

Aquest robot és una plataforma mòbil de 4 rodes. Ha estat modificat per enginyers de l'IRI degut a noves necessitats i el fi del suport per part de l'empresa fabricant. Els sensors rellevants pel nostre cas, marcats a la figura, són: el LIDAR, la IMU i els encoders de les rodes. Durant els experiments ha sigut teledirigit per un operador humà, i les dades han sigut recollides en un arxiu .bag a través de ROS.

6.1.1 Skid-steering

El sistema de direcció d'aquest robot és un "skid-steering", un sistema de direcció diferencial de 4 rodes. A diferència d'un cotxe normal, on les rodes del darrere son fixes i les del davant

poden virar, en un sistema "skid-steer" totes 4 rodes són fixes. Això implica un lliscament en qualsevol trajectòria no rectilínia. El gir s'aconsegueix aplicant diferents parells de força a les rodes dels dos costats. Per exemple, si les rodes de l'esquerra giren amb més força que les de la dreta ,el robot virarà cap a la dreta i viceversa. També existeix la possibilitat de girar sobre si mateix sense avançar, fent anar amb la mateixa força però sentits de gir contraris a les rodes dels dos costats.

Aquest sistema de direcció, des del punt de vista de l'estimació de l'estat, és problemàtic en els girs. La relació entre la diferència de forces en les rodes i la velocitat del gir depèn fortament de la fricció entre les rodes i el terra, ja que el gir es basa en lliscar sobre el terra. Això significa que el model que fem servir estarà fortament lligat al material del terra i les seves condicions (pols, humitat, etc). Ara bé, l'estimació relacionada al moviment rectilini serà molt acurada sempre i quan les rodes no llisquin al avançar. El robot compta amb un preprocessat de les dades provinents dels motors, proporcionant una odometria de rodes. Aquesta odometria és passada al processador de WOLF-core *processorOdom2d*.

6.1.2 LIDAR

El robot vé equipat amb un sensor LIDAR (Laser Imaging Detection And Ranging). El seu funcionament es basa en un feix de rajos làser que escombren l'espai a intervals determinats, donant-nos les distàncies fins als punts en els que incideixen.

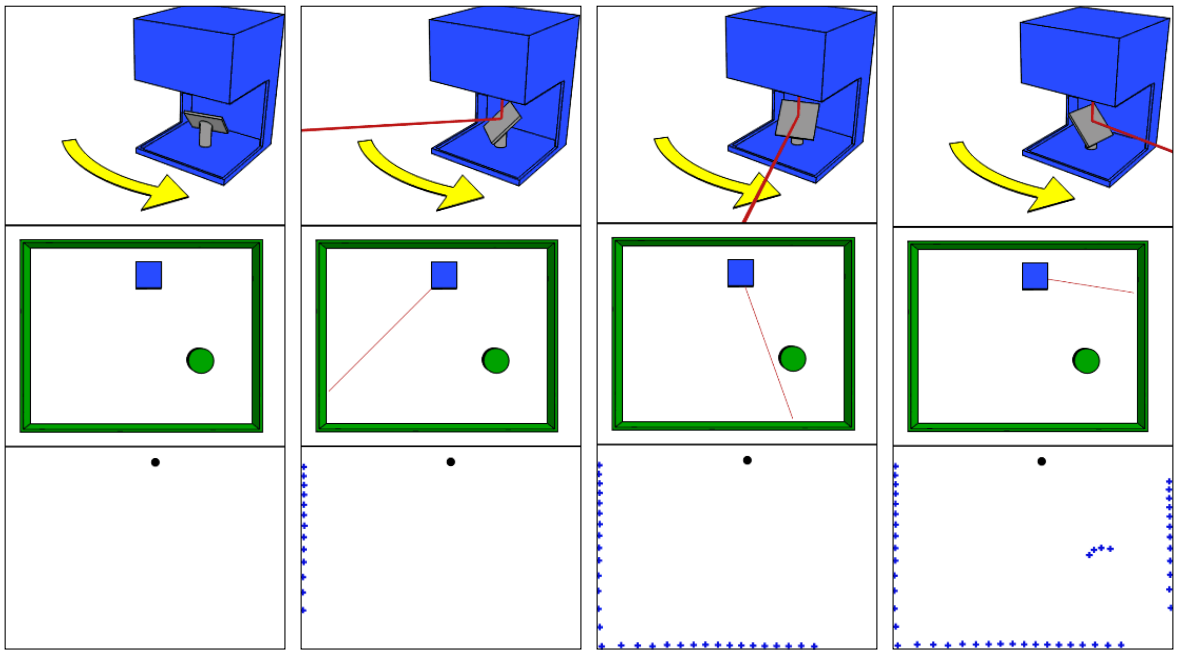

FIGURA 6.2: Diagrama simplificat del funcionament d'un LIDAR

.

Tot i que el sensor físic fa servir un feix de rajos làser per retornar un núvol de punts 3D, pel cas 2D simplement ens centrarem en el raig que escombra paral·lelament al terra. Aquest raig gira a 10 revolucions/s, i a cada 0.007 radians de gir ens diu la seva longitud (fins a un

màxim de 100m). Això ens permet reconstruir una aproximació visual de l'entorn del robot, i és el que farem servir visualment en els experiments per avaluar si l'estimació és acurada.

El plugin de laser té implementat, entre altres, el processador *processorOdomICP*. Aquest processador agafa els núvols de punts generats a dos keyframes consecutius i intenta alinear-los. Si ho aconsegueix (amb una certa tolerància d'error), podem saber quant s'ha mogut (i rotat) el robot entre aquests dos keyframes a partir de la diferència entre els núvols. Dels nostres 3 processadors, aquest és el més acurat donades condicions adients pel seu correcte funcionament (entorns "desendreçats" on els punts no es concentren tots en pocs conjunts).

6.1.3 IMU

El robot també ve equipat amb una IMU situada en una posició prou cèntrica. El més important, però, és que estigui en una part del robot fixa. Com que aquest robot no té actuadors (per exemple, braços), qualsevol lloc que no siguin les rodes seria adient. No entrarem en més detall, ja que ja hem parlat d'aquests sensors i del seu processador de WOLF al capítol 4.

6.2 Experiment: Petita volta al laboratori

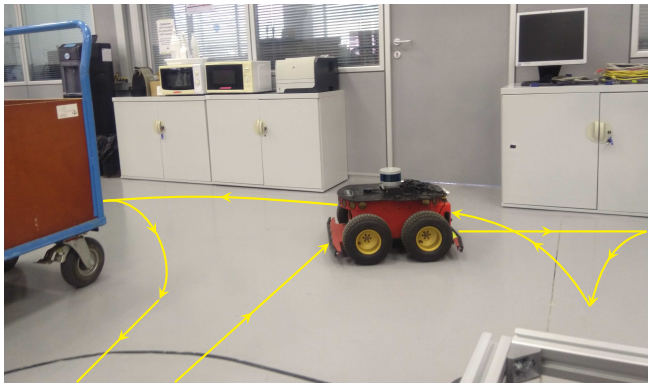

FIGURA 6.3: El laboratori amb la trajectòria seguida.

L'experiment és prou senzill, el robot segueix la trajectòria indicada en la imatge i recopila les dades dels sensors en l'arxiu .bag. Posteriorment, prenem aquest arxiu i el reproduïm amb ROS, donant-li les dades a WOLF amb 3 configuracions diferents per comparar els resultats. No s'inclou cap configuració on l'únic sensor és la IMU: a causa de la seva naturalesa, necessita tenir un altre sensor present per poder estimar el seu biaix. Sense aquesta estimació les dades serien inútils.

6.2.1 Configuració 1: LIDAR + Odometria de rodes + IMU

En aquesta configuració estan actius tots 3 sensors i els seus respectius processadors. L'entorn escollit és ideal pel LIDAR, que aconsegueix alinear els núvols de punts a gairebé tots

els keyframes. L'odometria de rodes i la IMU aporten informació extra que fa més acurada l'estimació, i permetrien l'estimació de paràmetres del LIDAR (per exemple, la posició relativa al robot). El mapa resultant és la nostra base de comparació ja que és el més acurat:

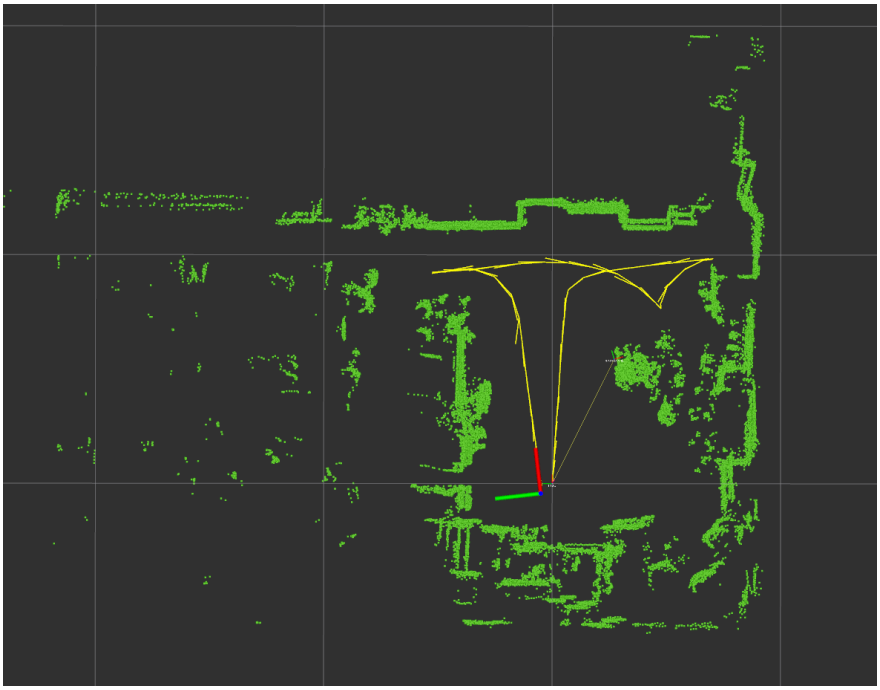

FIGURA 6.4: Mapa resultant de la configuració 1.

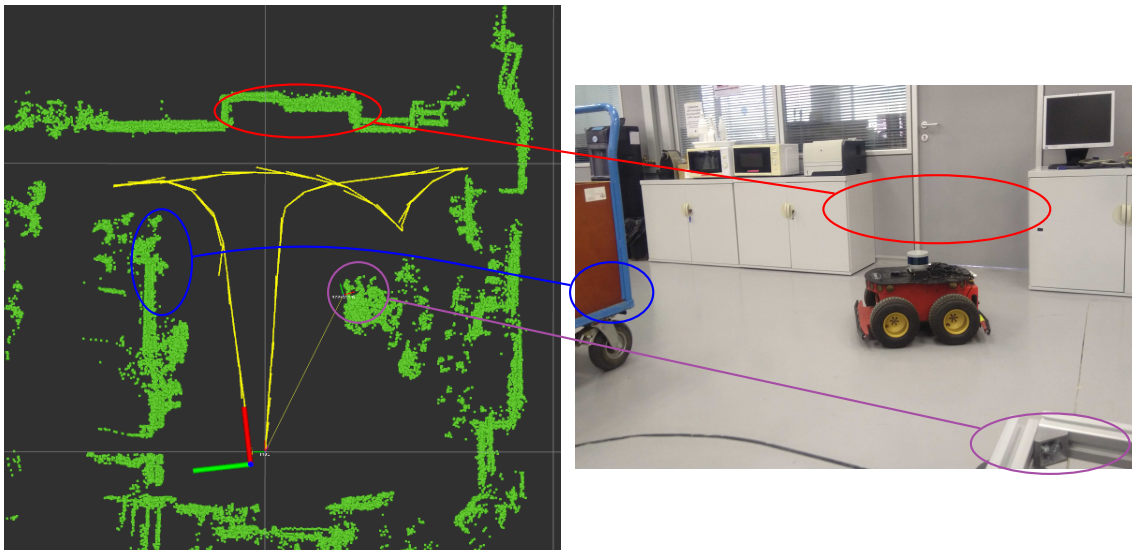

FIGURA 6.5: Comparació dels núvols de punts amb el laboratori.

6.2.2 Configuració 2: Odometria de rodes

En aquesta configuració l'únic sensor amb processador actiu és l'odometria de rodes. El LIDAR es manté actiu per qüestions de visualització, però està inactiu dins el graf. És a

dir, els únics factors que hi ha al graf són els corresponents a les rodes. Analitzarem els problemes mencionats abans sobre els girs pas a pas:

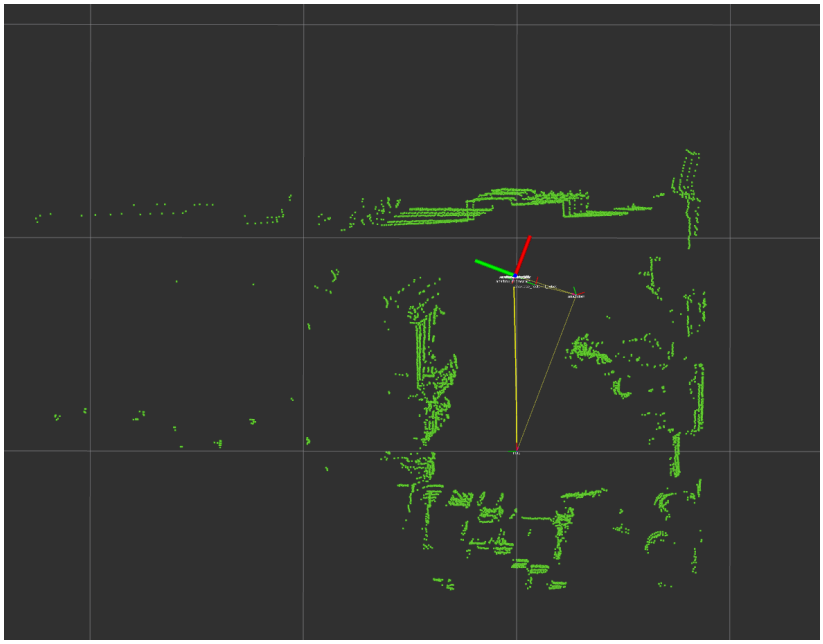

FIGURA 6.6: Moviment rectilini inicial.

L'estimació comença prou bé, com podem observar per l'agrupament del núvol de punts. Hi ha una lleugera tendència del robot de virar cap a l'esquerra, reflectida en com els núvols de punts estan separats. L'odometria de rodes, però, no ens diu que estigui girant cap a l'esquerra, ja que l'ordre que ha rebut el robot és la d'anar recte.[1]

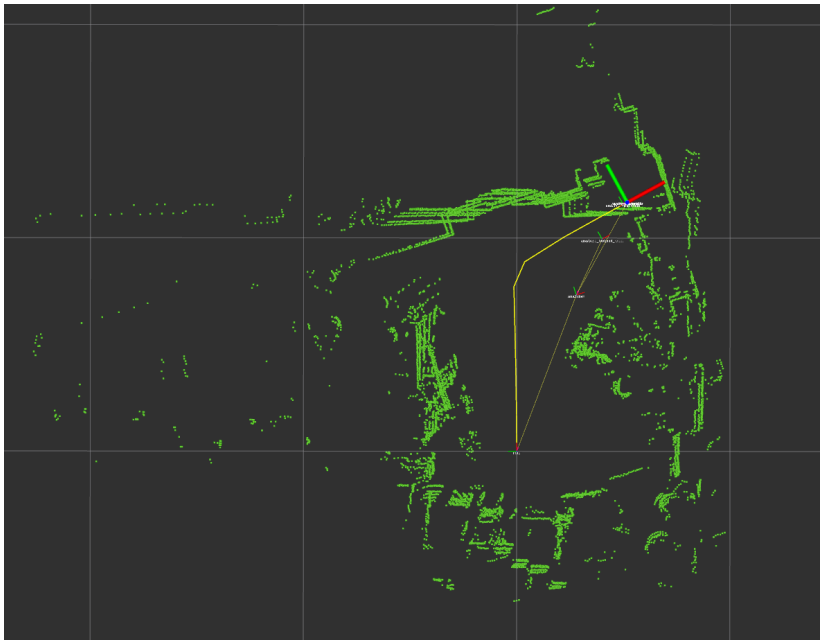

FIGURA 6.7: Primer gir.

[1] Aquesta tendència a virar lleugerament pot estar causada, per exemple, per una roda mal inflada.

En fer el primer gir observem els primers problemes greus en l'estimació. El model de l'odometria de rodes ens diu que hem girat un angle de més o menys 45°, però sabem que a la trajectòria real és de gairebé 90°. A causa d'això, els següents núvols de punts apareixen rotats 45° respecte on haurien de ser realment. A més a més, observem que la trajectòria creua una línia verda (que és un armariet).

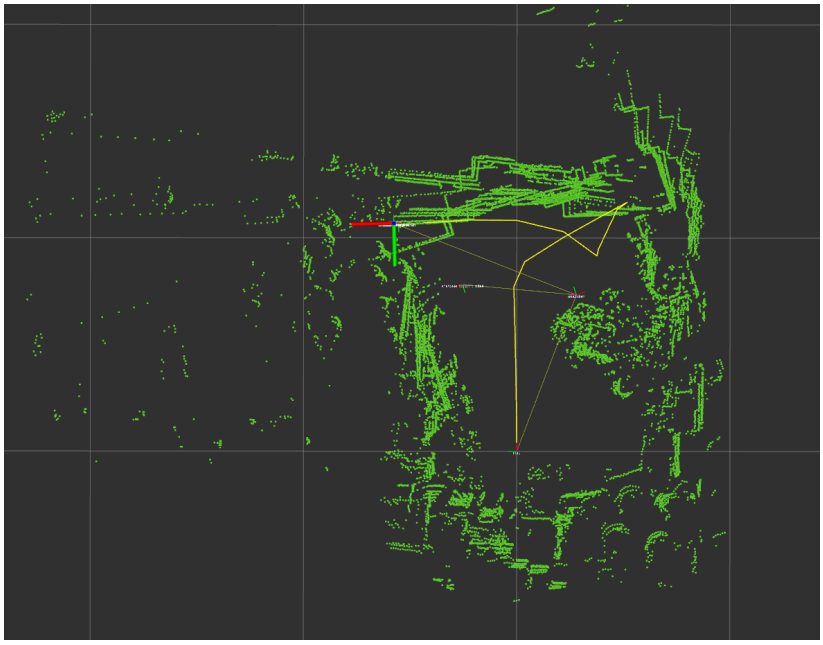

FIGURA 6.8: Segon moviment recte.

Com veiem, al fer marxa enrere i fer el segon gir aquest "compensa" l'error del primer gir, causant que els següents núvols de punts estiguin a uns -10° d'on haurien d'estar realment. Observem de nou que la trajectòria travessa ara una línia verda generada per l'error anterior.

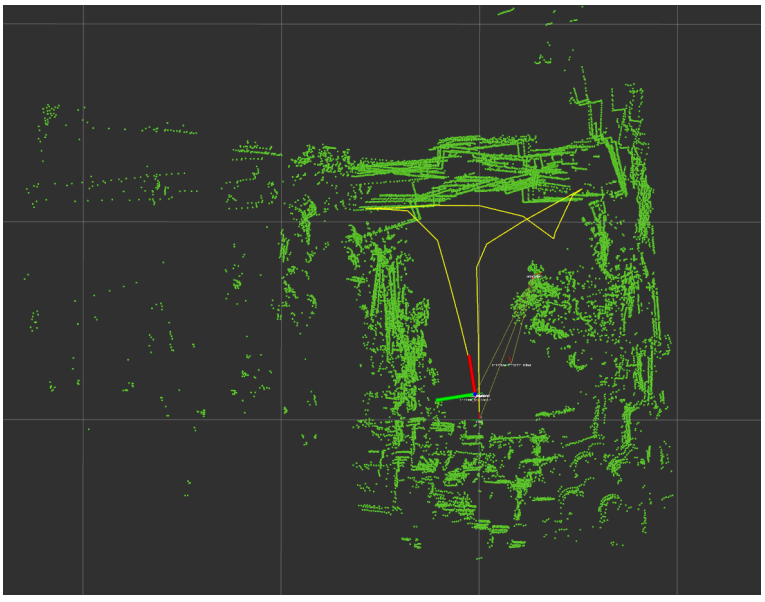

FIGURA 6.9: Mapa resultant de la configuració 2.

La trajectòria acaba assemblant-se molt a la descrita per la configuració 1, però el mapa està clarament malament.

6.2.3 Configuració 3: Odometria de rodes + IMU

En aquesta configuració afegim a la anterior el sensor i processador per la IMU. Al haver-hi ara més d'un sensor independent, comparant les mesures efectuades pels dos, el sistema pot estimar també els paràmetres de calibratge. Als arxius YAML de configuració li hem indicat a WOLF que la covariància de l'odometria de rodes en els girs és molt alta, causant que doni prioritat a les dades de gir provenint de la IMU.

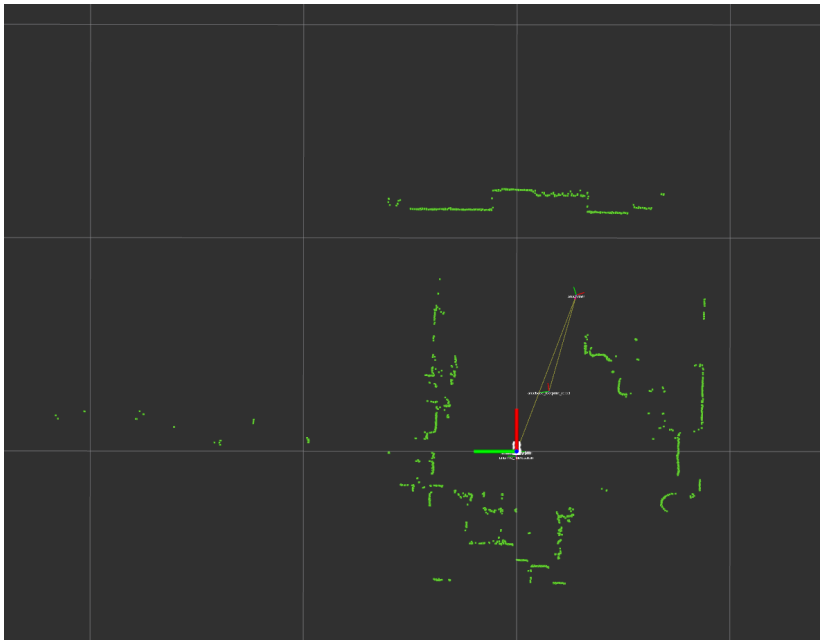

FIGURA 6.10: Situació inicial.

Durant els primers segons de l'experiment, el robot està quiet a la seva posició inicial. Això permet a WOLF fer una estimació inicial dels biaixos de la IMU bastant acurada, comparant l'odometria de rodes (que diu que està quiet) amb les dades de la IMU (que pel biaix diuen que estem en moviment).

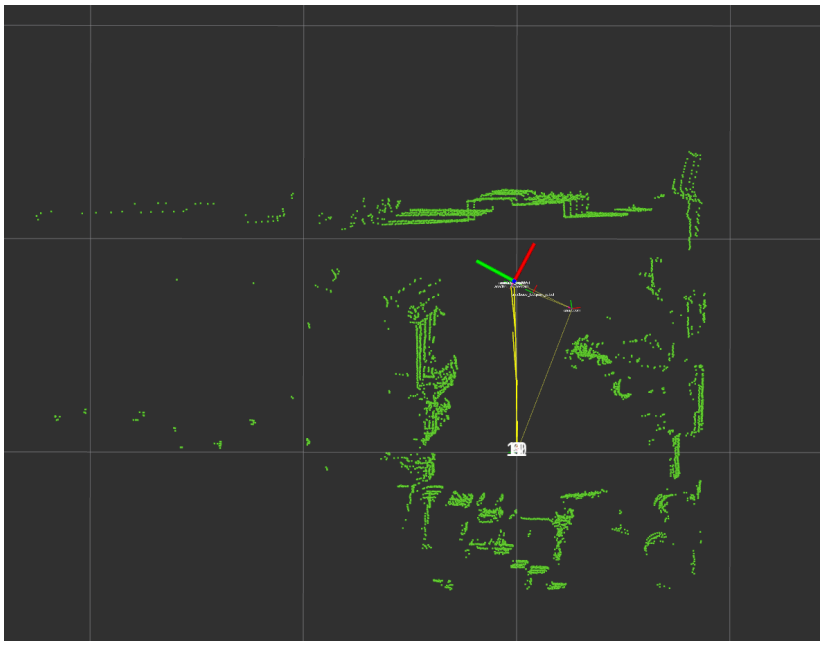

FIGURA 6.11: Primer moviment recte.

Durant el primer moviment recte, els resultats són pràcticament idèntics, ja que l'odometria de rodes pel moviment recte és bona i no discrepa amb el que diu la IMU.

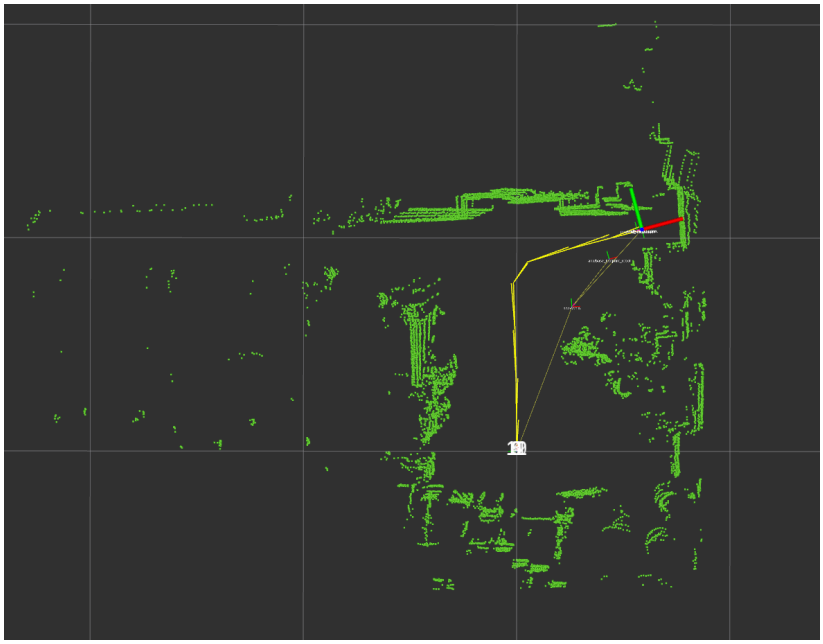

FIGURA 6.12: Primer gir.

En canvi, després del primer gir veiem una clara millora. La IMU ha pogut corregir l'error que causava abans l'odometria i la trajectòria és consistent amb l'entorn, ja que no creua cap línia verda.

Finalment, el resultat final té els núvols de punts millor posicionats, per tant l'estimació de la trajectòria és clarament millor que a la configuració 2.

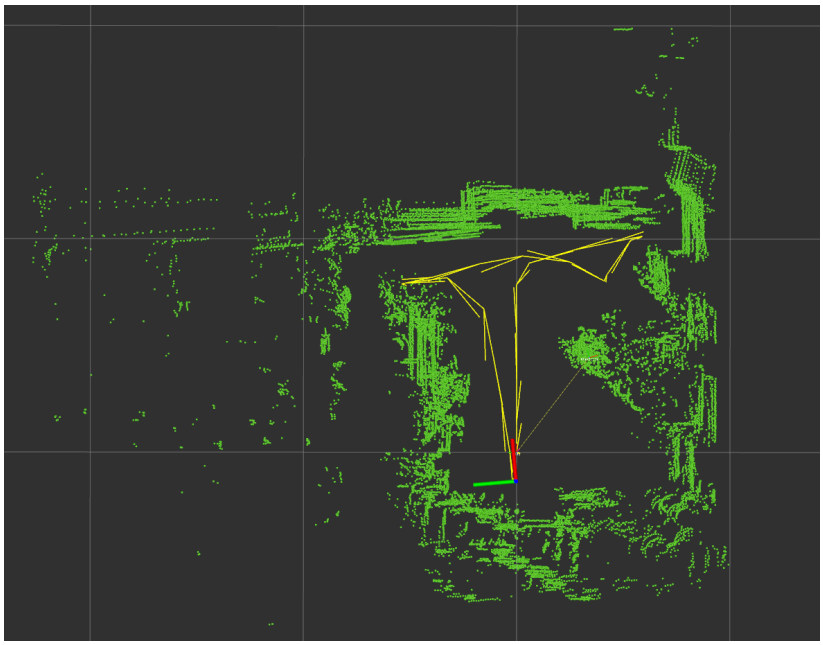

FIGURA 6.13: Mapa resultant de la configuració 3.

7 Conclusió

Al llarg d'aquest treball hem desenvolupat la teoria de preintegració en el cas particular de la IMU en un entorn 2D.

En el capítol 2 hem fet una introducció a SLAM, centrant-nos en la representació com a graf de factors. Hem explicat com el problema acaba reduït a optimitzar una funció no lineal que anomenem funció de cost, que és una suma d'errors amb pes.

Hem fet també una petita introducció a la teoria de Lie al capítol 3, centrant-nos en els seus aspectes rellevants en la posterior derivació de les deltes. La teoria de Lie és molt més extensa i complicada que el que hem exposat, però creiem que és un bon primer tast per qui vulgui entendre com formalitzar aquest tipus de problemes d'optimització en varietats.

La principal contribució d'aquest treball ha estat en el capítol 4, on hem desenvolupat la teoria de preintegració. Aquesta teoria no havia estat desenvolupada encara pel cas 2D, només pel cas 3D. Hi hem arribat a unes definicions de les deltes, que ens permeten actualitzar l'estat quan canvia l'estimació d'alguna keyframe, sense haver de reintegrar totes les dades. De manera similar, quan canvia l'estimació del biaix podem actualitzar les deltes sense haver de reintegrar. Això presenta una gran millora en el cost computacional.

Finalment, hem implementat tot això en un plugin de WOLF, que expliquem al capítol 5. Un cop implementat hem dut a terme uns experiments, descrits a capítol 6, on hem vist una clara millora en l'estimació de l'estat quan la IMU s'afegia a un sistema amb només odometria de rodes.

7.1 Experiments pendents

Tenim pendents encara una sèrie d'experiments. Per exemple, la IMU hauria de ser capaç de corregir una trajectòria del robot en què hi ha hagut un lliscament de les rodes amb el terra. L'odometria de rodes seria incapaç de detectar que això ha passat, però les dades de la IMU podrien reconstruir aquest lliscament.

Un altre experiment pendent és el de veure si amb la IMU podem evitar un fenomen relacionat amb l'estimació del LIDAR: quan els núvols de punts estan concentrats en dues línies rectes paral·leles (per exemple, quan el robot es mou per un passadís recte) pot passar que dos núvols de punts consecutius encaixin prou bé d'entrada (excepte molt pocs punts a les puntes dels núvols), causant que el robot pensi que està quiet quan en realitat està avançant. La IMU

podria ser capaç de corregir aquesta discrepància informant al LIDAR del moviment, donant lloc a un encaix més acurat que sí descrigui moviment.

7.2 Futures línies d'investigació

Com hem dit al capítol 4, no hem pogut fer servir les definicions de les Jacobianes corresponents al grup de Lie de les IMUs en forma compacta, degut a que no hem trobat la forma tancada de la jacobiana per la dreta de la varietat. Trobar aquesta expressió és complicat, però ens donaria molta més precisió en la propagació de covariàncies i actualitzacions de les deltes.

En passar de la descripció de les deltes en un espai 3D en la secció 4.3.2, a la dels espais 2D en la secció 4.3.4, hem projectat sobre el pla XY eliminant així la dependència de la gravetat g. Això introdueix la suposició que els eixos de la IMU i el pla del terra estan perfectament alineats, així com que el terra no té cap inclinació. Investigarem a partir d'aquí quins efectes tindria no suposar-ho, és a dir treballar amb superfícies inclinades.

Un cop estudiat això, es pot anar un pas més enllà i preguntar-nos què passaria amb superfícies on la inclinació canvia segons la posició. Duta al límit, aquesta situació seria la d'un robot mòbil movent-se sobre un terreny variable, com podria ser un rover en un altre planeta o un cotxe conduint per carreteres amb inclinació variable. Això seria molt útil en generalitzar el problema 2D i poder-lo aplicar a terrenys no controlats.

A Regles de derivació en grups de Lie

En tot aquest apèndix, a no ser que es digui el contrari, estarem usant les versions per la dreta dels operadors i les Jacobianes.

A.1 Regla de la cadena

Sigui $\mathcal{Y} = f(\mathcal{X})$ i $\mathcal{Z} = g(\mathcal{Y})$. La regla de la cadena és:

$$\frac{D\mathcal{Z}}{D\mathcal{X}} = \frac{D\mathcal{Z}}{D\mathcal{Y}}\frac{D\mathcal{Y}}{D\mathcal{X}} \tag{A.1}$$

o en forma de jacobianes:

$$\mathbf{J}^{\mathcal{Z}}_{\mathcal{X}} = \mathbf{J}^{\mathcal{Z}}_{\mathcal{Y}}\mathbf{J}^{\mathcal{Y}}_{\mathcal{X}} \tag{A.2}$$

Per demostrar-la aplicarem (3.21) tres vegades:

$$\left.\begin{array}{ccc} g(f(\mathcal{X}\oplus\boldsymbol{\tau})) & \underset{\boldsymbol{\tau}\to 0}{\longrightarrow} & g(f(\mathcal{X}))\oplus\mathbf{J}^{\mathcal{Z}}_{\mathcal{X}}\boldsymbol{\tau} \\ \boldsymbol{\tau}\downarrow & & \\ 0\downarrow & & \\ g(f(\mathcal{X})\oplus\mathbf{J}^{\mathcal{Y}}_{\mathcal{X}}\boldsymbol{\tau}) & \underset{\boldsymbol{\tau}\to 0}{\longrightarrow} & g(f(\mathcal{X}))\oplus\mathbf{J}^{\mathcal{Z}}_{\mathcal{Y}}\mathbf{J}^{\mathcal{Y}}_{\mathcal{X}}\boldsymbol{\tau} \end{array}\right\} \Longrightarrow \mathbf{J}^{\mathcal{Z}}_{\mathcal{X}} = \mathbf{J}^{\mathcal{Z}}_{\mathcal{Y}}\mathbf{J}^{\mathcal{Y}}_{\mathcal{X}} \tag{A.3}$$

A.2 Jacobiana de l'invers

$$\mathbf{J}^{\mathcal{X}^{-1}}_{\mathcal{X}} = \frac{D\mathcal{X}^{-1}}{D\mathcal{X}} = \lim_{\boldsymbol{\tau}\to 0}\frac{\mathrm{Log}((\mathcal{X}^{-1})^{-1}(\mathcal{X}\,\mathrm{Exp}(\boldsymbol{\tau}))^{-1})}{\boldsymbol{\tau}} \tag{A.4}$$

$$= \lim_{\boldsymbol{\tau}\to 0}\frac{\mathrm{Log}(\mathcal{X}\,\mathrm{Exp}(-\boldsymbol{\tau})\mathcal{X}^{-1})}{\boldsymbol{\tau}} \tag{A.5}$$

$$= \lim_{\boldsymbol{\tau}\to 0}\frac{(\mathcal{X}(-\boldsymbol{\tau})^{\wedge}\mathcal{X}^{-1})^{\vee}}{\boldsymbol{\tau}} \tag{A.6}$$

$$= -\mathbf{Ad}_{\mathcal{X}} \tag{A.7}$$

on hem usat (3.6) i (3.13).

A.3 Jacobiana de la composició

$$\mathbf{J}_{\mathcal{X}}^{\mathcal{X}\circ\mathcal{Y}} = \frac{D(\mathcal{X}\circ\mathcal{Y})}{D\mathcal{X}} = \lim_{\boldsymbol{\tau}\to 0} \frac{\mathrm{Log}((\mathcal{X}\circ\mathcal{Y})^{-1}\circ(\mathcal{X}\circ\mathrm{Exp}(\boldsymbol{\tau})\circ\mathcal{Y}))}{\boldsymbol{\tau}} \tag{A.8}$$

$$= \lim_{\boldsymbol{\tau}\to 0} \frac{\mathrm{Log}(\mathcal{Y}^{-1}\circ\mathrm{Exp}(\boldsymbol{\tau})\circ\mathcal{Y})}{\boldsymbol{\tau}} \tag{A.9}$$

$$= \lim_{\boldsymbol{\tau}\to 0} \frac{(\mathcal{Y}^{-1}(\boldsymbol{\tau})^{\wedge}\mathcal{Y})^{\vee})}{\boldsymbol{\tau}} \tag{A.10}$$

$$= \mathbf{Ad}_{\mathcal{Y}^{-1}} \tag{A.11}$$

$$= \mathbf{Ad}_{\mathcal{Y}^{-1}} \tag{A.12}$$

on hem usat (3.13) i (3.15).

De la mateixa manera:

$$\mathbf{J}_{\mathcal{Y}}^{\mathcal{X}\circ\mathcal{Y}} = \frac{D(\mathcal{X}\circ\mathcal{Y})}{D\mathcal{Y}} = \lim_{\boldsymbol{\tau}\to 0} \frac{\mathrm{Log}((\mathcal{X}\circ\mathcal{Y})^{-1}\circ(\mathcal{X}\circ\mathcal{Y}\circ\mathrm{Exp}(\boldsymbol{\tau})))}{\boldsymbol{\tau}} \tag{A.13}$$

$$= \lim_{\boldsymbol{\tau}\to 0} \frac{\boldsymbol{\tau}}{\boldsymbol{\tau}} \tag{A.14}$$

$$= I \tag{A.15}$$

A.4 Propietats de la jacobiana de la varietat

Sigui $\delta\boldsymbol{\tau}$ una petita pertorbació a $\mathbb{R}^m$. Aplicant la definició de jacobiana arribem a les següents aproximacions:

$$\mathrm{Exp}(\boldsymbol{\tau}+\delta\boldsymbol{\tau}) \approx \mathrm{Exp}(\boldsymbol{\tau})\,\mathrm{Exp}(\mathbf{J}_r(\boldsymbol{\tau})\delta\boldsymbol{\tau}) \tag{A.16}$$

$$\mathrm{Exp}(\boldsymbol{\tau})\,\mathrm{Exp}(\delta\boldsymbol{\tau}) \approx \mathrm{Exp}(\boldsymbol{\tau}+\mathbf{J}_r(\boldsymbol{\tau})\delta\boldsymbol{\tau}) \tag{A.17}$$

$$\mathrm{Log}(\mathrm{Exp}(\boldsymbol{\tau})\,\mathrm{Exp}(\delta\boldsymbol{\tau})) \approx \boldsymbol{\tau}+\mathbf{J}_r^{-1}(\boldsymbol{\tau})\delta\boldsymbol{\tau} \tag{A.18}$$

Podem també definir la jacobiana de la varietat per l'esquerra com:

$$\mathbf{J}_l(\boldsymbol{\tau}) := \frac{^{\mathcal{E}}D\,\mathrm{Exp}(\boldsymbol{\tau})}{D\boldsymbol{\tau}} \tag{A.19}$$

amb aproximacions anàlogues:

$$\mathrm{Exp}(\boldsymbol{\tau}+\delta\boldsymbol{\tau}) \approx \mathrm{Exp}(\mathbf{J}_l(\boldsymbol{\tau})\delta\boldsymbol{\tau})\,\mathrm{Exp}(\boldsymbol{\tau}) \tag{A.20}$$

$$\mathrm{Exp}(\delta\boldsymbol{\tau})\,\mathrm{Exp}(\boldsymbol{\tau}) \approx \mathrm{Exp}(\boldsymbol{\tau}+\mathbf{J}_l(\boldsymbol{\tau})\delta\boldsymbol{\tau}) \tag{A.21}$$

$$\mathrm{Log}(\mathrm{Exp}(\delta\boldsymbol{\tau})\,\mathrm{Exp}(\boldsymbol{\tau})) \approx \boldsymbol{\tau}+\mathbf{J}_l^{-1}(\boldsymbol{\tau})\delta\boldsymbol{\tau} \tag{A.22}$$

A.5 Jacobiana de Log()

A partir de A.18:

$$\mathbf{J}_{\mathcal{X}}^{\mathrm{Log}(\mathcal{X})} = \mathbf{J}_r^{-1}(\mathrm{Log}(\mathcal{X})) \tag{A.23}$$

A.6 Jacobiana dels operadors suma i resta

$$\mathbf{J}_{\mathcal{X}}^{\mathcal{X}\oplus\boldsymbol{\tau}} = \mathbf{J}^{\mathcal{X}\circ\mathrm{Exp}(\boldsymbol{\tau})} = \mathbf{Ad}_{\mathrm{Exp}(\boldsymbol{\tau})^{-1}} \tag{A.24}$$

$$\mathbf{J}_{\boldsymbol{\tau}}^{\mathcal{X}\oplus\boldsymbol{\tau}} = \mathbf{J}_{\mathrm{Exp}(\boldsymbol{\tau})}^{\mathcal{X}\circ\mathrm{Exp}(\boldsymbol{\tau})}\mathbf{J}_{\boldsymbol{\tau}}^{\mathrm{Exp}(\boldsymbol{\tau})} = \mathbf{J}_r(\boldsymbol{\tau}) \tag{A.25}$$

$$\mathbf{J}_{\mathcal{X}}^{\mathcal{X}\ominus\mathcal{X}} = \mathbf{J}_{\mathcal{X}^{-1}\circ\mathcal{Y}}^{\mathrm{Log}(\mathcal{X}^{-1}\circ\mathcal{Y})}\mathbf{J}_{\mathcal{X}^{-1}}^{\mathcal{X}^{-1}\circ\mathcal{Y}}\mathbf{J}_{\mathcal{X}}^{\mathcal{X}^{-1}} = -\mathbf{J}_l^{-1}(\boldsymbol{\tau}) \tag{A.26}$$

$$\mathbf{J}_{\mathcal{Y}}^{\mathcal{X}\ominus\mathcal{X}} = \mathbf{J}_{\mathcal{X}^{-1}\circ\mathcal{Y}}^{\mathrm{Log}(\mathcal{X}^{-1}\circ\mathcal{Y})}\mathbf{J}_{\mathcal{Y}}^{\mathcal{X}^{-1}\circ\mathcal{Y}} = \mathbf{J}_r^{-1}(\boldsymbol{\tau}) \tag{A.27}$$

B Desenvolupaments del capítol 4

B.1 Preintegració incremental de les dades

Substituint la integració discreta de (4.12):

$$\begin{aligned}\mathbf{p}_k &= \mathbf{p}_j + \mathbf{v}_j\delta t + \frac{1}{2}\mathbf{g}\delta t^2 + \frac{1}{2}\mathbf{R}_j\mathbf{a}_j\delta t^2\\ \mathbf{v}_k &= \mathbf{v}_j + \mathbf{g}\delta t + \mathbf{R}_j\mathbf{a}_j\delta t\\ \mathbf{R}_k &= \mathbf{R}_j\exp\left([\boldsymbol{\omega}_j\delta t]_\times\right)\end{aligned} \tag{B.1}$$

en la definició de les deltes (4.13):

$$\begin{aligned}\Delta\mathbf{p}_{ij} &= \mathbf{R}_i^\top\left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2\right)\\ \Delta\mathbf{v}_{ij} &= \mathbf{R}_i^\top\left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right)\\ \Delta\mathbf{R}_{ij} &= \mathbf{R}_i^\top\mathbf{R}_j\end{aligned} \tag{B.2}$$

arribem a:

$$\begin{aligned}\Delta\mathbf{p}_{ik} &= \mathbf{R}_i^\top\left(\mathbf{p}_j + \mathbf{v}_j\delta t + \frac{1}{2}\mathbf{g}\delta t^2 + \frac{1}{2}\mathbf{R}_j\mathbf{a}_j\delta t^2 - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ik} - \frac{1}{2}\mathbf{g}\Delta t_{ik}^2\right)\\ &= \mathbf{R}_i^\top\left(\mathbf{p}_j - \mathbf{p}_i + \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2 + (\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij})\delta t + \frac{1}{2}\mathbf{R}_j\mathbf{a}_j\delta t^2\right)\\ &= \Delta\mathbf{p}_{ij} + \Delta\mathbf{v}_{ij}\delta t + \frac{1}{2}\Delta\mathbf{R}_{ij}\mathbf{a}_j\delta t^2\\ \Delta\mathbf{v}_{ik} &= \Delta\mathbf{R}_i^\top\left(\mathbf{v}_j + \mathbf{g}\delta t + \mathbf{R}_i\mathbf{a}_j\delta t - \mathbf{v}_i - \mathbf{g}\Delta t_{ik}\right)\\ &= \mathbf{R}_i^\top\left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij} + \mathbf{g}\delta t - \mathbf{g}\delta t + \mathbf{R}_i\mathbf{a}_j\delta t\right)\\ &= \Delta\mathbf{v}_{ij} + \Delta\mathbf{R}_{ij}\mathbf{a}_j\delta t\\ \Delta\mathbf{R}_{ik} &= \mathbf{R}_i^\top\mathbf{R}_j\exp\left([\boldsymbol{\omega}_j\delta t]_\times\right)\\ &= \Delta\mathbf{R}_{ij}\exp\left([\boldsymbol{\omega}_j\delta t]_\times\right)\end{aligned} \tag{B.3}$$

on hem usat que $\Delta t_{ik} = \delta t + \Delta t_{ij}$.

B.2 L'espai tangent

$$\Delta^{-1}\Delta = \Delta_\varepsilon \iff \frac{\partial}{\partial\xi}\left(\Delta^{-1}\Delta\right) = \frac{\partial}{\partial\xi}\left(\Delta_\varepsilon\right) = 0 \iff \dot{\Delta}^{-1}\Delta + \Delta^{-1}\dot{\Delta} = 0 \tag{B.4}$$

Trobant que

$$\dot{\Delta^{-1}} = \left[\begin{array}{c|c|c} \Delta\dot{\mathbf{R}}^\top & -\left(\Delta\dot{\mathbf{R}}^\top\Delta\mathbf{v} + \Delta\mathbf{R}^\top\Delta\dot{\mathbf{v}}\right) & \begin{array}{c}-\Delta\dot{\mathbf{R}}^\top\left(\Delta\mathbf{p} - \Delta\mathbf{v}\Delta t\right)\\ -\Delta\mathbf{R}^\top\left(\dot{\Delta\mathbf{p}} - \dot{\Delta\mathbf{v}}\Delta t - \Delta\mathbf{v}\dot{\Delta t}\right)\end{array} \\ \hline 0 & 1 & -\dot{\Delta t} \\ \hline 0 & 0 & 1 \end{array}\right] \tag{B.5}$$

i definint les variacions instantànies $\mathfrak{v} =: \frac{\partial\Delta\mathbf{p}}{\partial\xi}$, $\mathfrak{a} := \frac{\partial\Delta\mathbf{v}}{\partial\xi}$, $\mathfrak{s} := \frac{\partial\Delta t}{\partial\xi}$, i la matriu $[\omega] := \Delta\mathbf{R}^\top\dot{\Delta\mathbf{R}}$ tenim:

$$\Delta^{-1}\dot{\Delta} = \begin{bmatrix} [\omega] & \Delta\mathbf{R}^\top\mathfrak{a} & \Delta\mathbf{R}^\top(\mathfrak{v} - \mathfrak{s}\Delta\mathbf{v}) \\ 0 & 0 & \mathfrak{s} \\ 0 & 0 & 0 \end{bmatrix} \tag{B.6}$$

$$-\dot{\Delta^{-1}}\Delta = \begin{bmatrix} -\Delta\dot{\mathbf{R}}^\top\Delta\mathbf{R} & \Delta\mathbf{R}^\top\mathfrak{a} & \Delta\mathbf{R}^\top(\mathfrak{v} - \mathfrak{s}\Delta\mathbf{v}) \\ 0 & 0 & \mathfrak{s} \\ 0 & 0 & 0 \end{bmatrix} \tag{B.7}$$

És a dir:

$$\Delta^{-1}\dot{\Delta} + \dot{\Delta^{-1}}\Delta = \begin{bmatrix} [\omega] + [\omega]^\top & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = 0 \iff [\omega] \text{ és antisimetrica} \tag{B.8}$$

Per tant l'estructura de l'element $\boldsymbol{\tau}^\wedge = \Delta^{-1}\dot{\Delta}$ és la d'una delta on la $\Delta\mathbf{R}$ és una matriu antisimètrica i la resta són paràmetres lliures.

B.3 Expressions tancades de $\mathcal{R}$, $\mathcal{Q}$, $\mathcal{P}$

En el nostre cas de dimensió 2 tenim les següents propietats per $[\theta]_\times$:

$$\begin{aligned} [1]_\times &= \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \\ [\theta]_\times &= \begin{bmatrix} 0 & -\theta \\ \theta & 0 \end{bmatrix} = \theta\,[1]_\times \\ [\theta]_\times^2 &= -\theta^2 I \\ [\theta]_\times^3 &= \theta^3\,[1]_\times = -\theta^2\,[\theta]_\times \\ [\theta]_\times^4 &= \theta^4 I \end{aligned} \tag{B.9}$$

Usant aquest fet, $\mathcal{R}, \mathcal{Q}$ i $\mathcal{P}$ esdevenen:

$$\begin{aligned}
\mathcal{R} &= I + [\theta]_\times + \frac{1}{2}[\theta]_\times^2 + \frac{1}{3!}[\theta]_\times^3 + \dots \\
&= I + [\theta]_\times - \frac{\theta^2}{2} I - \frac{\theta^2}{3!}[\theta]_\times + \dots \\
&= I \cdot (1 - \frac{\theta^2}{2} + \frac{\theta^4}{4!} + \dots) + [\theta]_\times \cdot (1 - \frac{\theta^2}{3!} + \frac{\theta^4}{5!} + \dots) \\
&= \cos(\theta) I + \frac{\sin(\theta)}{\theta}[\theta]_\times \\
&= \cos(\theta) I + \sin(\theta)[1]_\times = \mathbf{R}(\theta)
\end{aligned} \tag{B.10}$$

$$\begin{aligned}
\mathcal{Q} &= I + \frac{1}{2}[\theta]_\times + \frac{1}{3!}[\theta]_\times^2 + \frac{1}{4!}[\theta]_\times^3 + \dots \\
&= I + \frac{1}{2}[\theta]_\times - \frac{\theta^2}{3!} I - \frac{\theta^2}{4!}[\theta]_\times + \dots \\
&= I(1 - \frac{\theta^2}{3!} + \frac{\theta^4}{5!} + \dots) + [\theta]_\times (\frac{1}{2} - \frac{\theta^2}{4!} + \frac{\theta^4}{6!} + \dots) \\
&= I\frac{\sin(\theta)}{\theta} + [\theta]_\times \frac{1 - \cos(\theta)}{\theta^2} \\
&= I\frac{\sin(\theta)}{\theta} + [1]_\times \frac{1 - \cos(\theta)}{\theta}
\end{aligned} \tag{B.11}$$

$$\begin{aligned}
\mathcal{P} &= \frac{1}{2} I + \frac{1}{3!}[\theta]_\times + \frac{1}{4!}[\theta]_\times^2 + \frac{1}{5!}[\theta]_\times^3 + \dots \\
&= \frac{1}{2} I + \frac{1}{3!}[\theta]_\times - \frac{\theta^2}{4!} I - \frac{\theta^2}{5!}[\theta]_\times + \dots \\
&= I(\frac{1}{2} - \frac{\theta^2}{4!} + \frac{\theta^4}{6!} + \dots) + [\theta]_\times (\frac{1}{3!} - \frac{\theta^2}{5!} + \frac{\theta^4}{7!} + \dots) \\
&= I\frac{1 - \cos(\theta)}{\theta^2} + [\theta]_\times \frac{\theta - \sin(\theta)}{\theta^3} \\
&= I\frac{1 - \cos(\theta)}{\theta^2} + [1]_\times \frac{\theta - \sin(\theta)}{\theta^2}
\end{aligned} \tag{B.12}$$

A partir d'aquí és trivial trobar les seves derivades:

$$\frac{d\mathcal{R}(\theta)}{d\theta} = -\sin(\theta) I + \cos(\theta)[1]_\times = \mathcal{R}(\theta)[1]_\times \tag{B.13}$$

$$\frac{d\mathcal{Q}(\theta)}{d\theta} = \frac{\theta\cos(\theta) - \sin(\theta)}{\theta^2} I + \frac{\theta\sin(\theta) + \cos(\theta) - 1}{\theta^2}[1]_\times \tag{B.14}$$

$$\frac{d\mathcal{P}(\theta)}{d\theta} = \frac{\theta\sin(\theta) + 2\cos(\theta) - 2}{\theta^3} I - \frac{\theta - 2\sin(\theta) + \theta\cos(\theta)}{\theta^3}[1]_\times \tag{B.15}$$

C Arxius

A continuació segueix una llista dels arxius creats/modificats en la creació del plugin per la IMU en 2D:

Creats

1. `include/imu/factor/factor_imu2d.h`
2. `test/gtest_factor_imu2d.cpp`
3. `include/imu/math/imu2d_tools.h`
4. `test/gtest_imu2d_tools.cpp`
5. `include/imu/processor/processor_imu2d.h`
6. `src/processor/processor_imu2d.cpp`
7. `test/processor_imu2d_UnitTester.cpp`
8. `test/processor_imu2d_UnitTester.h`
9. `test/gtest_processor_imu2d.cpp`
10. `demos/processor_imu2d.yaml`
11. `src/yaml/processor_imu2d_yaml.cpp`
12. `include/imu/sensor/sensor_imu2d.h`
13. `src/sensor/sensor_imu2d.cpp`
14. `demos/sensor_imu2d.yaml`
15. `src/yaml/sensor_imu2d_yaml.cpp`
16. `include/imu/feature/feature_imu2d.h`
17. `src/feature/feature_imu2d.cpp`
18. `CMakeLists.txt`
19. `test/CMakeLists.txt`

Modificats

1. `include/imu/capture/capture_imu.h`
2. `src/capture/capture_imu.cpp`

3. `src/sensor/sensor_imu.cpp`
4. `test/gtest_feature_imu.cpp`
5. `test/gtest_processor_imu.cpp`

Bibliografia

D. Atchuthan. *Towards new sensing capabilities for legged locomotion using real-time state estimation with low-cost IMUs*. Theses, Université Paul Sabatier - Toulouse III, Oct. 2018. URL https://tel.archives-ouvertes.fr/tel-02088756.

C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. On-manifold preintegration theory for fast and accurate visual-inertial navigation. *CoRR*, abs/1512.02363, 2015. URL http://arxiv.org/abs/1512.02363.

M. Fourmy, D. Atchuthan, N. Mansard, J. Solà, and T. Flayols. Absolute humanoid localization and mapping based on imu lie group and fiducial markers. In *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, pages 237–243, 2019.

T. Lupton and S. Sukkarieh. Efficient integration of inertial observations into visual slam without initialization. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1547–1552, 2009.

M. Quigley, J. Faust, T. Foote, J. Leibs, et al. Ros: an open-source robot operating system.

J. Solà. Generalized sensor self-calibration in graph-slam (internal document), 2019.

J. Solà, J. Deray, and D. Atchuthan. A micro lie theory for state estimation in robotics, 2020.

J. Solà*, J. Vallvé-Navarro, J. Casals, J. Deray, M. Fourmy, D. Atchuthan, and J. Andrade-Cetto. Wolf: a modular estimation framework for robotics based on factor graphs. Article sense publicar, 2021.