

UNIVERSIDAD NACIONAL DEL LITORAL



Algoritmos de Navegación y Guiado para Vehículos Autónomos Basados en Métodos de Horizonte Móvil

Guido Marcelo Sánchez

FICH

FACULTAD DE INGENIERIA
Y CIENCIAS HIDRICAS

INTEC

INSTITUTO DE DESARROLLO TECNOLÓGICO
PARA LA INDUSTRIA QUIMICA

Tesis de Doctorado **2017**



UNIVERSIDAD NACIONAL DEL LITORAL
Facultad de Ingeniería y Ciencias Hídricas
Instituto de Desarrollo Tecnológico para la Industria Química

ALGORITMOS DE NAVEGACIÓN Y GUIADO PARA VEHÍCULOS AUTÓNOMOS BASADOS EN MÉTODOS DE HORIZONTE MÓVIL

Guido Marcelo Sánchez

Tesis remitida al Comité Académico del Doctorado
como parte de los requisitos para la obtención
del grado de
DOCTOR EN INGENIERIA
Mención Inteligencia Computacional, Señales y Sistemas
de la
UNIVERSIDAD NACIONAL DEL LITORAL

2017

Comisión de Posgrado, Facultad de Ingeniería y Ciencias Hídricas, Ciudad Universitaria, Paraje "El Pozo",
S3000, Santa Fe, Argentina.



UNIVERSIDAD NACIONAL DEL LITORAL
Facultad de Ingeniería y Ciencias Hídricas
Instituto de Desarrollo Tecnológico para la Industria Química

ALGORITMOS DE NAVEGACIÓN Y GUIADO PARA VEHÍCULOS AUTÓNOMOS BASADOS EN MÉTODOS DE HORIZONTE MÓVIL

Guido Marcelo Sánchez

Lugar de trabajo:

Instituto de Investigación en Señales, Sistemas e Inteligencia
Computacional – sinc(*i*)
UNL – CONICET

Director:

Dr. Leonardo Giovanini sinc(*i*)
Universidad Nacional del Litoral
CONICET

Co-director:

Dr. Alejandro Limache CIMEC
Universidad Nacional del Litoral
CONICET

Jurado Evaluador:

Dr. Alejandro Gonzalez INTEC
Universidad Nacional del Litoral
CONICET

Dr. Ernesto Kofman CIFASIS
Universidad Nacional de Rosario
CONICET

Dr. Vicente Mut INAUT
Universidad Nacional de San Juan
CONICET

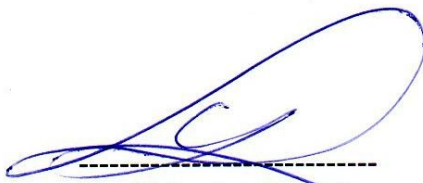


UNIVERSIDAD NACIONAL DEL LITORAL
Facultad de Ingeniería y Ciencias Hídricas

Santa Fe, 21 de marzo de 2017.

Como miembros del Jurado Evaluador de la Tesis de Doctorado en Ingeniería titulada *“Algoritmos de navegación y guiado para vehículos autónomos basados en métodos de horizonte móvil”*, desarrollada por el Ing. Guido Marcelo SANCHEZ, en el marco de la Mención “Inteligencia Computacional, Señales y Sistemas”, certificamos que hemos evaluado la Tesis y recomendamos que sea aceptada como parte de los requisitos para la obtención del título de Doctor en Ingeniería.

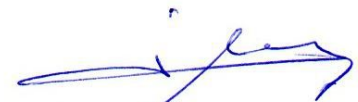
La aprobación final de esta disertación estará condicionada a la presentación de dos copias encuadernadas de la versión final de la Tesis ante el Comité Académico del Doctorado en Ingeniería.



Dr. Alejandro Gonzalez



Dr. Ernesto Kofman



Dr. JOSE LUIS MACOR
SECRETARIO DE POSGRADO
Facultad de Ingeniería y Cs. Hídricas

Dr. Vicente Mut (v)

Santa Fe, 21 de marzo de 2017.

Certifico haber leído la Tesis, preparada bajo mi dirección en el marco de la Mención “Inteligencia Computacional, Señales y Sistemas” y recomiendo que sea aceptada como parte de los requisitos para la obtención del título de Doctor en Ingeniería.

.....
Dr. Alejandro Limache
Codirector de Tesis

.....
Dr. Leonardo Giovanini
Director de Tesis

(*) Participo por videoconferencia.



Dr. JOSE LUIS MACOR
SECRETARIO DE POSGRADO
Facultad de Ingeniería y Cs. Hídricas

Universidad Nacional del Litoral
Facultad de Ingeniería y
Ciencias Hídricas

Secretaría de Posgrado

Ciudad Universitaria
C.C. 217
Ruta Nacional N° 168 - Km. 472,4
(3000) Santa Fe
Tel: (54) (0342) 4575 229
Fax: (54) (0342) 4575 224
E-mail: posgrado@fich.unl.edu.ar

DECLARACION DEL AUTOR

Esta tesis ha sido remitida como parte de los requisitos para la obtención del grado académico de Doctor en Ingeniería - Mención Inteligencia Computacional, Señales y Sistemas ante la Universidad Nacional del Litoral y ha sido depositada en la Biblioteca de la Facultad de Ingeniería y Ciencias Hídricas para que esté a disposición de sus lectores bajo las condiciones estipuladas por el Reglamento de la mencionada Biblioteca.

Citaciones breves de esta tesis son permitidas sin la necesidad de un permiso especial, en la suposición de que la fuente sea correctamente citada. Solicitudes de permiso para la citación extendida o para la reproducción parcial o total de este manuscrito serán concebidos por el portador legal del derecho de propiedad intelectual de la obra.

Agradecimientos

A mi director, Dr. Leonardo Giovanini, por brindarme la oportunidad de entrar al mundo de la investigación. Todo su tiempo dedicado en guiarme ha permitido que alcance este objetivo.

A los compañeros del laboratorio y de oficina. No podría haber tenido un mejor lugar de trabajo que el $\text{sync}(i)$.

A mi familia y en especial a mis padres por su ejemplo y apoyo incondicional. Gracias por todo lo que han hecho por mí.

A Sheila, por estar a mi lado en las buenas y en las malas, su cariño, paciencia y tolerancia. Recorrer este camino fue más fácil gracias a ella.

Finalmente, quiero agradecer a las siguientes instituciones:

- $\text{sync}(i)$: Instituto de Investigación en Señales, Sistemas e Inteligencia Computacional.
- Facultad de Ingeniería y Ciencias Hídricas, Universidad Nacional del Litoral.
- CONICET: Consejo Nacional de Investigaciones Científicas y Técnicas.

Guido M. Sánchez
Santa Fe, noviembre de 2016.

Índice general

1. Introducción	1
1.1. Organización de la Tesis	4
1.2. Resultados	5
2. Modelado cinemático de robots móviles terrestres	7
2.1. Introducción	7
2.2. Representación de la posición del robot	9
2.2.1. Marcos de referencia	10
2.2.2. Vectores, proyecciones y matrices de cosenos directores	11
2.2.3. Matrices de rotación	15
2.2.4. Dinámica de las matrices de transformación	22
2.3. Cinemática de los robots móviles	24
2.3.1. Teorema del transporte	25
2.3.2. Transformación de velocidades	26
2.3.3. Ecuación de una rueda	27
2.4. Tracción diferencial	28
2.4.1. Tracción diferencial en 2D	30
2.4.2. Tracción diferencial en 3D	31
2.5. Resultados	34
2.6. Conclusiones	34
3. Navegación	39
3.1. Introducción	39
3.2. Estimación basada en horizonte móvil lineal	43
3.3. Actualización del costo de arribo	45
3.4. Análisis de estabilidad	51
3.4.1. Estabilidad acotada	53
3.5. Resultados	53
3.6. Estimación basada en horizonte móvil no lineal	57
3.7. Resultados	59
3.8. Navegación con MHE	64

3.9. Resultados	67
3.10. Conclusiones	71
4. Guiado	73
4.1. Introducción	73
4.2. Control predictivo no lineal	75
4.3. Control predictivo no lineal robusto	79
4.4. Control predictivo no lineal robusto iterado	81
4.5. Resultados	83
4.6. Conclusiones	86
5. Múltiples vehículos	89
5.1. Introducción	89
5.2. Control Predictivo Distribuido	91
5.2.1. Control predictivo	91
5.2.2. MPC distribuido	92
5.3. Propiedades	96
5.3.1. Performance	96
5.3.2. Convergencia	99
5.3.3. Factibilidad	102
5.3.4. Estabilidad	103
5.4. Comportamiento bajo fallas de comunicación	103
5.5. Resultados	106
5.6. Conclusiones	107
6. Conclusiones y trabajos futuros	111
6.1. Conclusiones	111
6.2. Trabajos futuros	112
Apéndice A. Demostraciones del Capítulo 3	115
A.1. Demostración de la Observación 3.3	115
A.2. Demostración de la Observación 3.4	115
A.3. Demostración del Teorema 3.1	116
Apéndice B. Demostraciones del Capítulo 4	117
B.1. Demostración del Teorema 4.1	117
B.2. Demostración del Teorema 4.3	119
Apéndice C. Demostraciones del Capítulo 5	121
C.1. Demostración del Lema 5.1	121
C.2. Demostración del Teorema 5.1	122
C.3. Demostración del Teorema 5.2	124

Índice de figuras

2.1. Proyección de \mathbf{s}_1 en \mathcal{R} .	12
2.2. Rotación de un vector \mathbf{a} sobre el eje \mathbf{u} a una nueva posición \mathbf{d} .	16
2.3. Secuencia de rotaciones correspondientes a cada eje coordenado.	19
2.4. Marcos de referencia en movimiento: Dos observadores que se encuentran en movimiento relativo miden el movimiento del objeto.	26
2.5. Marcos de referencia para la cinemática de una rueda.	27
2.6. Lego NXT configurado como un robot móvil de tracción diferencial.	29
2.7. Tracción diferencial. En esta configuración hay dos ruedas montadas sobre un mismo eje.	30
2.8. Tracción diferencial en 3D. Se agrega la rueda del soporte al modelo para tener en cuenta el ángulo de rotación θ sobre el eje y .	32
2.9. Simulación en terreno plano.	35
2.10. Simulación en presencia de una rampa inclinada.	36
3.1. Esquema de actualización de MHE.	41
3.2. Comparación de los estimadores para el modelo (3.16) con horizonte de $N = 5$.	56
3.3. Comparación de los estimadores para el modelo (3.16) con horizonte de $N = 5$ y σ_w variable.	56
3.4. Evolución de las secuencias de factores de olvido para MHE_{VF} y MHE_{CT} .	57
3.5. Comparación de los estimadores para el modelo (3.19) con horizonte de $N = 5$.	62
3.6. Comparación de los estimadores para el modelo (3.19) con horizonte de $N = 5$ y σ_w variable.	63
3.7. Estimación de la posición p^e y la velocidad v^e .	68
3.8. Comparación del cuaternión estimado \hat{q}_b^e y el real q_b^e .	69
3.9. Estimación de los <i>bias</i> α y β .	69

4.1. Esquema del funcionamiento de MPC.	73
4.2. Posición y controles aplicados en ENU.	84
4.3. Estimación de la posición p y el cuaternión q en ENU.	85
4.4. Tiempo de ejecución para MHE y MPC con horizonte $N = 5$.	86
5.1. Visualización de las condiciones de optimalidad generadas por las Ecuaciones (5.13).	98
5.2. Posiciones de los vehículos utilizando a) MPC centralizado y b) MPC descentralizado.	108
5.3. Posiciones y acciones de control de los vehículos utilizando MPC centralizado y MPC descentralizado.	109
5.4. Distancia entre los robots obtenidas por MPC centralizado y descentralizado.	110

Índice de tablas

3.1. Comparación del efecto del tamaño del horizonte de estimación en la sumatoria del error cuadrático medio del estado $x^{(1)}$ para el sistema (3.16).	54
3.2. Comparación del efecto del tamaño del horizonte de estimación en la sumatoria del error cuadrático medio del estado $x^{(2)}$ para el sistema (3.16).	55
3.3. Comparación del efecto del tamaño del horizonte de estimación en la sumatoria del error cuadrático medio del estado $x^{(1)}$ para el sistema (3.19).	60
3.4. Comparación del efecto del tamaño del horizonte de estimación en la sumatoria del error cuadrático medio del estado $x^{(2)}$ para el sistema (3.19).	61

Resumen

El objetivo de esta tesis es desarrollar algoritmos de navegación y guiado para vehículos autónomos terrestres. Para ello, son necesarios dos componentes esenciales: *i*) un sistema de estimación, que le permita saber en qué lugar del espacio se encuentra y su orientación, y *ii*) un sistema de control, que le permita controlar la orientación y posición del vehículo.

En el caso del problema de estimación, se calculan los estimados de los estados y parámetros del sistema a partir de mediciones con ruido obtenidas de los sensores disponibles en el vehículo, mientras que en el caso del problema de control, se calculan las acciones de control que debe ejecutar cada uno de los actuadores para seguir una trayectoria predefinida. Los límites físicos en el comportamiento de los vehículos, los sensores y los actuadores pueden ser modelados como restricciones. Por este motivo, las técnicas de estimación y control capaces de incorporar restricciones han adquirido un nuevo auge.

Los métodos de horizonte móvil se basan en la solución de un problema de optimización explícito a cada instante de muestreo. Por este motivo, son capaces de incorporar la información de las restricciones del sistema en la etapa de diseño y de esta forma, proveen un marco teórico unificado para la estimación y el control de sistemas con restricciones.

En esta tesis se han estudiado tres temas principales: *i*) Estimación de horizonte móvil con restricciones. Esta técnica está basada en mínimos cuadrados y presenta mejoras significativas con respecto a la utilización del filtro de Kalman y del filtro de Kalman Extendido. A fin de poder ser implementada en sistemas de tiempo real, se deben utilizar horizontes de tamaño reducido. Esto se logra mediante una buena aproximación del *costo de arribo*. Se propuso utilizar una aproximación adaptativa del costo de arribo utilizando algoritmos de olvido variable y una propagación recursiva de la matriz de peso correspondiente. El método propuesto es capaz de obtener buenas aproximaciones del costo de arribo con bajo costo computacional. *ii*) Control predictivo para sistemas no lineales, basado en la linealización de sistemas no lineales a lo largo de una trayectoria predefinida. Esta técnica nos permite controlar el modelo de un vehículo sin necesidad de desacoplarlo en subsis-

temas y se ha estudiado la estabilidad y robustez del algoritmo. *iii)* Control de múltiples vehículos utilizando algoritmos de control distribuido basados en juegos dinámicos, con agentes que resuelven el problema descentralizado equivalente al problema centralizado en un entorno de red. Estos agentes pueden cooperar y comunicarse entre ellos para alcanzar los objetivos del sistema completo.

Los aportes de esta Tesis de Doctorado consisten en el método adaptativo para la aproximación del costo de arribo descrito en el Capítulo [3](#); la demostración de estabilidad a lazo cerrado de la técnica de control predictivo no lineal robusta basada en la restricción contractiva descrita en el Capítulo [4](#); y la demostración de equivalencia entre el controlador centralizado y el algoritmo de control predictivo distribuido basado en juegos dinámicos descrito en el Capítulo [5](#).

Palabras clave: Métodos de horizonte móvil, Estimación con restricciones, Control con restricciones, Sistemas multiagente, Teoría de juegos.

Abstract

The objective of this thesis is to develop navigation and guidance algorithms for autonomous terrestrial vehicles. To do this, two essential components are required: *i*) an estimation system, which allows to know the position and orientation of the vehicle, and *ii*) a control system, which allows to control the position and orientation of the vehicle.

In the case of the estimation problem, we calculate the estimates of the states and parameters of the system from noisy measurements obtained from the sensors available in the vehicle, while in the case of the control problem, we calculate the control actions which must run each of the actuators to follow a predefined path. Physical limits on the behaviour of vehicles, sensors and actuators can be modelled as constraints. For this reason, the estimation and control techniques capable of incorporating restrictions have gained a lot of attention.

Moving horizon methods are based on the solution of an explicit optimization problem at every sample instant. For this reason, they are able to incorporate system constraint information into the design stage and thus provide a unified theoretical framework for estimation and control of constrained systems.

In this thesis, three main topics have been studied: *i*) Moving horizon estimation with restrictions. This technique is based on least squares and presents significant improvements regarding the use of the Kalman filter and the Extended Kalman filter. In order to be implemented in real-time systems, small horizons must be used. This is achieved by a good approximation of the arrival cost. We propose to use an adaptive approximation of the arrival cost using variable forgetting algorithms and a recursive propagation of the corresponding weight matrix. The proposed method is able to obtain good approximations of the cost of arrival with low computational cost. *ii*) Model predictive control for nonlinear systems, based on the linearization of nonlinear systems along a predefined path. This technique allows us to control the model of a vehicle without having to decouple it in subsystems and the stability and robustness of the algorithm are studied. *iii*) Control of multi-

ple vehicles using distributed control algorithms based on dynamic games, with agents that solve the decentralized problem equivalent to the centralized problem in a network environment. These agents can cooperate and communicate with each other to achieve the objectives of the whole system.

The contributions of this Doctoral Thesis consist of the adaptive method for the approximation of the cost of arrival described in Chapter 3; the closed loop stability demonstration of the robust nonlinear predictive control technique based on the contractive constraint described in Chapter 4; and the demonstration of equivalence between the centralized controller and the distributed predictive control algorithm based on dynamic games described in Chapter 5.

Keywords: Moving horizon methods, Constrained estimation, Constrained control, Multiagent systems, Game theory.

Capítulo 1

Introducción

Para que un vehículo pueda moverse de manera autónoma en el espacio, depende de dos componentes esenciales: *i*) un sistema de estimación, que le permita saber en qué lugar del espacio se encuentra y su orientación, y *ii*) un sistema de control, que le permita controlar la orientación y posición del vehículo. Es por ello que el estudio y desarrollo de técnicas de estimación y control capaces de incorporar restricciones han adquirido un nuevo auge en el ámbito académico de la robótica.

La mayoría de los algoritmos de control utilizados en el guiado y navegación de vehículos suponen, en la etapa de diseño, que los estados y parámetros del sistema están disponibles al momento de calcular las acciones de control. En la práctica, sin embargo, la única información disponible son mediciones con ruido. Por este motivo, los estados y parámetros del vehículo deben ser determinados a partir de estas mediciones utilizando un modelo dinámico. Este problema ha sido estudiado desde 1960, obteniéndose un sinnúmero de métodos basados en diferentes medidas estadísticas, siendo el más popular el filtro de Kalman [21, 51].

Los límites físicos en el comportamiento de los vehículos y sus trayectorias suelen ser modelados a través de restricciones. La omisión de esta información en los algoritmos de estimación y control puede degradar sustancialmente sus desempeños [45]. Desafortunadamente, el filtro de Kalman no permite incluir explícitamente restricciones en los estados o parámetros en su formulación. Por este motivo se han desarrollado métodos ad-hoc que permiten agregar restricciones, tales como reducción del modelo [117], proyección de la estimación [44], proyección de la ganancia [121], truncado de la función de densidad de probabilidad [118] y proyección del sistema [58]. Estos métodos dan como resultado soluciones sub-óptimas en el mejor de los casos.

Las técnicas de estimación de horizonte móvil (Moving Horizon Estimation – MHE) se basan en la solución de un problema de optimización para

encontrar una estimación de los estados y parámetros en cada instante de tiempo [38, 95]. De esta forma, MHE provee un marco teórico unificado para la estimación con restricciones dado que resuelve un problema de horizonte finito para determinar los estados y parámetros del sistema. A medida que llegan nuevas mediciones, las viejas se descartan de la ventana de estimación y se vuelve a resolver el problema de estimación para determinar la estimación de estados y parámetros actuales (ver Fig. 3.1). La información de las mediciones que fueron descartadas se asimilan en la función objetivo como un término extra que recibe el nombre *costo de arribo*. El *costo de arribo* es un término de la función objetivo que caracteriza la distribución estadística de los estados y parámetros dada la información provista por las mediciones desde el comienzo del proceso de estimación hasta una muestra atrás. De esta forma, MHE aproxima al problema de mínimos cuadrados *batch* (estimador de información completa) mientras mantiene una ventana de datos reducida al considerar una cantidad finita de mediciones (ventana de estimación).

Rao et al. [97] propusieron la actualización del costo de arribo mediante el uso de un filtro de Kalman. La suposición gaussiana empleada por estos métodos [96, 97], conduce a aproximaciones inadecuadas del costo de arribo. Este problema surge del hecho que la presencia de restricciones en los estados y parámetros modifican la distribuciones de probabilidad de los ruidos y destruye la hipótesis de independencia estadística entre los estimados y los ruidos [102]. Es por ello que se han propuesto métodos alternativos para actualizar el costo de arribo, que utilizan filtros no lineales basados en muestreo [70, 124]. Si bien estos enfoques obtienen buenos resultados, presentan el problema de una excesiva carga computacional que los hace problemático para la aplicación en guiado y navegación de vehículos.

Por otro lado, las técnicas de control basadas en horizonte móvil (Model Predictive Control – MPC) resuelven un problema de optimización sobre una ventana de predicción. De esta forma, se calculan las acciones de control mediante la minimización de una función objetivo sujeta a restricciones. Una vez obtenidas las acciones de control, se aplica la primera de ellas y la ventana avanza una muestra hacia adelante [72, 99]. En la última década, ha habido un auge en la aplicación de MPC a sistemas no lineales. En general este interés ha coincidido con el desarrollo de algoritmos de optimización no lineales (Non Linear Programs – NLP) más eficientes y la evolución de la capacidad de cálculo de las nuevas generaciones de microprocesadores [11, 31, 65, 69]. Sin embargo, este tipo de soluciones generalmente son computacionalmente demandantes y además las restricciones impuestas por las dinámicas no lineales generalmente conllevan a plantear NLP no convexos. Esto conduce a que sólo se pueda aplicar a sistemas dinámicos con períodos de muestreos grandes. Es por ello que la búsqueda de algoritmos de MPC para sistemas

no lineales continúa siendo un área de investigación activa.

Una de estas líneas de trabajo utiliza técnicas de linealización y aproximación lineal para lidiar con el problema de la complejidad computacional asociada a los NLP [12, 80]. La principal ventaja de estos métodos radica en el hecho que el modelo utilizado para la predicción es un conjunto lineal de aproximaciones locales, convirtiendo de esta manera, el NLP en un conjunto convexo de problemas de lineales (Linear Program – LP) [12, 25, 67] y permiten abordar aspectos de robustez. Sin embargo, estas técnicas no aseguran de forma automática la estabilidad a lazo cerrado. Este problema ha sido estudiado en [79] y [80]. Una de las maneras de resolver el problema de la estabilidad es agregar una restricción al problema de optimización que garantice la contracción de los estados a lo largo del tiempo. Esta idea fue originalmente propuesta por Yang y Polack [132] y la demostración de estabilidad fue desarrollada por De Olivera y Morari [27]. En este trabajo se impuso esta restricción de contracción sobre los estados, lo cual es muy restrictivo en términos computacionales y limita la cantidad de soluciones posibles de la dinámica a lazo cerrado resultante de la optimización. Para relajar estos problemas Murillo, Sánchez y Giovanini [84] propusieron un algoritmo que sólo impone la restricción de contractividad a la función de costo. De este modo permite una mayor cantidad de soluciones, a la vez que se limita el espacio de búsqueda del programa de optimización lo cual redundaría en una mayor eficiencia computacional.

Hasta ahora hemos hablado de estimar y controlar un solo vehículo. Al momento de considerar dos o más vehículos estamos presentes ante lo que se denomina un sistema multiagente. El control de sistemas multiagente puede verse como una capa superior que controla y establece trayectorias a la capa de navegación y guiado de un solo vehículo antes detallados. Tradicionalmente, dos paradigmas predominan en el control de sistemas multiagente: *i*) la utilización de un controlador centralizado, o *ii*) la utilización de un conjunto de controladores descentralizados.

Si bien los controladores centralizados a lazo cerrado presentan un rendimiento superior, hay muchas razones por las que el control descentralizado es el método dominante, pues proporciona flexibilidad en la operación, escalabilidad, tolerancia a fallos, diseño más simple, entre otras ventajas [40]. El análisis y diseño de sistemas multiagente tiene como objetivo principal obtener un comportamiento colectivo, o solución global, a partir del diseño del comportamiento individual de cada agente. Bajo ciertas condiciones se puede utilizar juegos de potencial o acíclicos para modelar soluciones al problema de control de sistemas multiagente [75].

Mientras que los algoritmos de planeamiento de misión existentes emplean un enfoque centralizado [108, 112], el algoritmo propuesto por Giovanini et

al. [40] emplea un enfoque distribuido basado en teoría de juegos [87] en el que el problema de planeamiento es resuelto dividiéndolo en un conjunto de subproblemas que son distribuidos entre los agentes de la misión. También se propone un esquema de particionado para el problema de planeamiento que reduce las interacciones y dependencia entre los subproblemas. Cuando los subproblemas son resueltos utilizando el enfoque propuesto, producen una solución óptima global al problema de planeamiento estratégico. Sin embargo, cada subproblema puede ser resuelto independientemente, en cuyo caso se obtiene una solución subóptima. Esta característica brinda robustez operacional y autonomía al sistema.

Marden et al. [74, 75] modelaron las interacciones del sistema multiagente como un juego no cooperativo, expresando el comportamiento colectivo deseado como un equilibrio de Nash. Los autores hicieron énfasis en algoritmos de aprendizaje simples que garanticen convergencia a un equilibrio y en requerimientos computacionales mínimos para poder utilizar dicha implementación en una amplia gama de problemas. De esta manera, establecieron una conexión entre la clase de juegos de potencial y el control cooperativo, demostrando los beneficios de esta conexión en varios problemas de control cooperativo tales como: cobertura dinámica de sensoramiento, problemas de consenso y enrutamiento distribuido en una red.

Cabe destacar que los enfoques propuestos por Giovanini et al. [40] y Marden et al. [74, 75] son equivalentes en sus fundamentos, solo difieren en la metodología utilizada para resolver el juego resultante: Giovanini et al. [40] utilizan un problema de optimización explícita mientras que Marden et al. [74, 75] utilizan métodos basados en aprendizaje.

En este contexto, el trabajo de Sánchez et al. [106] se centra en resolver el problema de control distribuido para sistemas LTI desde la perspectiva de teoría de juegos. En este enfoque, el problema centralizado se transforma en un juego dinámico que resuelve varios problemas de optimización locales, que intercambian entre ellos la información relevante de las variables de decisión de cada subsistema. De esta forma, se reduce la carga computacional y se evitan los obstáculos propios de implementaciones centralizadas mientras que se mantienen sus propiedades (estabilidad, optimalidad, factibilidad).

1.1. Organización de la Tesis

Esta Tesis está organizada en dos temas: *i)* navegación y *ii)* guiado. Cada uno de estos temas será abordado en un capítulo específico y se incluirá un capítulo sobre técnicas de modelado de robots móviles terrestres.

El Capítulo 2 trata el modelado cinemático de robots móviles terrestres

y analiza las técnicas utilizadas a tal fin. Aquí se plantea la necesidad de una técnica generalizada que permita modelar de forma sencilla y sistemática robots móviles de distintas características geométricas. La utilización del método de cadenas cinemáticas permite definir la velocidad y aceleración de los vehículos en términos de ecuaciones diferenciales algebraicas y evita aplicar rotaciones hasta una vez conformado el modelo, lo que permite obtener de forma sencilla.

En el Capítulo 3 se presenta un algoritmo de estimación basado en horizonte móvil aplicado en sistemas lineales y no lineales. Se describe la equivalencia entre mínimos cuadrados y la estimación de horizonte móvil y se introduce el término **costo de arribo**, el cual se encarga de resumir la información previa al instante inicial del proceso de estimación. Se muestra que mediante una buena aproximación del costo de arribo, se puede reducir el tamaño del horizonte manteniendo una buena performance.

El Capítulo 4 presenta el problema de guiado para vehículos autónomos. En primer instancia, se presenta el algoritmo iterativo propuesto y luego se lo aplica en el problema de guiado para el caso de un robot de tracción diferencial en el plano.

En el Capítulo 5 se trata el control de múltiples vehículos de forma descentralizada, mostrando la equivalencia del esquema centralizado con un juego de potencial, donde los vehículos coordinan sus decisiones.

Finalmente, en el Capítulo 6 se presentan las conclusiones del trabajo realizado, analizando sus ventajas y desventajas respecto a otros enfoques y se presentan las líneas de trabajo futuras a explorar.

1.2. Resultados

En el área de estimación se han estudiado métodos de estimación basados en horizonte móvil, lo cual presenta un avance con respecto al estándar de facto que es el filtro de Kalman y sus variantes. Asimismo, se ha propuesto un método para aproximar el costo de arribo, el cual permite reducir el horizonte de estimación y posibilita la operación en tiempo real. Finalmente, se utilizó MHE para fusionar –mediante simulación– datos de un sensor de posicionamiento, acelerómetro, giróscopo y magnetómetro en los marcos de referencia *Earth Centered Earth Fixed* y *East North Up*, logrando así mejorar la estimación de posición mientras se obtiene de forma indirecta la orientación del vehículo en el marco de referencia elegido.

En el área de control, se han estudiado técnicas de control predictivo robusto para sistemas no lineales. El problema no lineal se resuelve mediante una representación lineal variante en el tiempo y se resuelve un problema

de optimización lineal cuadrático equivalente, donde se garantiza la robustez, estabilidad y convergencia. Este algoritmo fue utilizado para resolver los problemas de guiado de forma satisfactoria. De esta forma, todo indica que el algoritmo propuesto puede conseguir una respuesta similar a los algoritmos clásicos utilizados en control predictivo no lineal, pero con menor carga computacional.

Finalmente, con el objetivo de estudiar el control de múltiples vehículos (o flotas) de forma descentralizada, se propone un algoritmo de control distribuido basado en juegos dinámicos, donde cada uno de los vehículos resuelve su problema de control de forma independiente y coordina sus acciones con los demás vehículos a través de una red de comunicaciones con el fin de alcanzar el objetivo global. De esta forma, si somos capaces de controlar un vehículo, seremos capaces de controlar una flota utilizando este algoritmo.

Estos resultados han sido comunicados a través de los siguientes trabajos:

- Sánchez, G., Murillo, M., & Giovanini, L. (2017). Adaptive arrival cost update for improving Moving Horizon Estimation performance. *ISA Transactions*, 68, 54-62. <http://doi.org/10.1016/j.isatra.2017.02.012>.
- Murillo, M., Sánchez, G., & Giovanini, L. (2016). Iterated non-linear model predictive control based on tubes and contractive constraints. *ISA Transactions*, 62, 120–128. <http://doi.org/10.1016/j.isatra.2016.01.008>.
- Sánchez, G., Giovanini, L., Murillo, M., & Limache, A. (2011). Distributed Model Predictive Control Based on Dynamic Games. In *Advanced Model Predictive Control*. InTech. <http://doi.org/10.5772/16268>.

Capítulo 2

Modelado cinemático de robots móviles terrestres

2.1. Introducción

El proceso necesario para entender el movimiento de un robot móvil terrestre (WMR, Wheeled Mobile Robot) comienza con el estudio de la contribución que realiza cada rueda al movimiento. La interacción de cada rueda del vehículo con el terreno es lo que permite el movimiento y a la vez, impone restricciones al movimiento, por ejemplo, no permitiendo desplazamientos laterales [114]. Por lo tanto, comandar un WMR es una tarea más compleja que controlar las velocidades de dos o más ruedas unidas mediante ejes a un cuerpo rígido. Por ello, el desarrollo de modelos matemáticos precisos es de sumo interés para los procesos de estimación, control y planeamiento de trayectoria de los WMR.

Los modelos de movimiento pueden variar en su complejidad. Pueden ser **cinemáticos**, en los cuales se calculan las velocidades del vehículo y se integran para obtener su posición y actitud (u orientación); o pueden ser **dinámicos**, en los cuales se calculan las aceleraciones y se integran dos veces para calcular la posición y actitud. La mayoría de los modelos cinemáticos desarrollados tienen en cuenta el caso del movimiento en dos dimensiones (2D), ya que son fáciles de obtener y de bajo costo computacional. Debido a esto, mucho del esfuerzo de investigación se ha concentrado en robots simples que se desplazan por terreno plano [20, 33, 64, 113, 114]. La derivación de las ecuaciones cinemáticas para los WMR en 2D puede clasificarse en dos enfoques principales:

- **Enfoque geométrico:** utiliza las relaciones geométricas para determinar el modelo del WMR, encontrando las relaciones que determinan

el eje instantáneo de rotación del cuerpo rígido. Por este motivo, no es general y cada caso debe ser analizado de forma particular.

- **Enfoque basado en transformaciones:** consiste en aplicar una serie de transformaciones de coordenadas y sus derivadas para relacionar el movimiento de las ruedas con el movimiento del vehículo.

Una de las principales contribuciones al enfoque basado en transformaciones es el trabajo de Muir y Neuman [82], donde asignan marcos de referencia en un grafo de transformaciones que cumple con la convención de Sheth-Uicker. Estos modelos consideran desplazamientos en el plano x - y ángulo de rotación ψ sobre el eje z .

Para poder trasladarse por terreno arbitrario o realizar maniobras agresivas es necesario disponer de modelos más precisos y avanzados, donde las características del terreno sobre el que el WMR se desplaza puedan ser tenidas en cuenta. Los WMR con articulaciones y suspensiones son capaces de desplazarse por terreno arbitrario y adaptar su configuración a la topología cambiante. Están siendo utilizados para exploración planetaria, agricultura, minería, defensa y en ambientes de acceso peligroso. La incorporación de suspensiones actuadas hace que éstos vehículos sean capaces de modificar su estructura, cambiando así su centro de masa y posibilitando que se adapten a los terrenos más variados sin perder el equilibrio. Para cumplir con estas necesidades, necesitamos que la formulación de modelos para WMR sea:

- **Precisa:** que tenga en cuenta articulaciones, las características de los actuadores y la interacción con el terreno, incluyendo si alguna rueda pierde contacto.
- **General:** que permita derivar el modelo para cualquier configuración de WMR a partir de métodos sistemáticos.
- **Modular:** que permita agregar cualquier modelo de actuadores o de contacto con el terreno.
- **Rápida:** permita la simulación varios órdenes de magnitud más rápido que el tiempo real.

Varios trabajos recientes se enfocan en el problema del movimiento en tres dimensiones (3D) para la cinemática de los WMR en terreno arbitrario. Tarokh y McDermott [119, 120] desarrollaron una versión 3D del método basado en transformaciones de Muir y Neuman [82] y lo aplicaron con el modelo del rover Rocky7 para realizar simulaciones en terreno arbitrario. Mediante la definición de una cadena de transformaciones homogéneas que siguen la

convención Denavit–Hartenbert (DH), son capaces de estimar y controlar el movimiento del WMR en los seis grados de libertad (DOF, Degrees of Freedom) disponibles. Este enfoque es sistemático, metódico y trata el caso general de terreno arbitrario, deslizamiento en las ruedas y suspensiones.

Recientemente, los trabajos de Kelly y Seegmiller [55, 56, 57] proponen el uso de cadenas cinemáticas para definir la velocidad y aceleración de los vehículos en términos de ecuaciones diferenciales algebraicas, donde no es necesario (pero si posible) adherirse a la convención DH. Utilizan el *Teorema del Transporte* para desarrollar un modelo tridimensional genérico que describe el movimiento de las ruedas y a partir de ello, describen el movimiento de un vehículo articulado. La *ecuación de la rueda* es la base para las expresiones de la cinemáticas. Estos trabajos fueron los primeros en combinar un método general el desarrollo de modelos cinemáticos 3D y la predicción del movimiento utilizando ecuaciones diferenciales algebraicas (DAEs, Differential Algebraic Equations). Posteriormente, este enfoque fue mejorado para tener en cuenta modelo de contacto de la rueda con el terreno [109, 110, 111], de forma tal que son capaces de predecir los efectos de los deslizamientos de las ruedas en el vehículo a partir de su interacción con terrenos arbitrarios, en vez de reaccionar a ellos.

En este capítulo se presentará la notación que permitirá expresar el movimiento del WMR en distintos marcos de referencia. Utilizando esta notación, se construirán modelos cinemáticos para el caso de un WMR de tracción diferencial. Primero se estudiará el desarrollo del modelo para el caso 2D y luego para el caso 3D. En este último caso, se tendrá en cuenta la interacción del vehículo con superficies arbitrarias bajo la suposición de que no hay deslizamiento de las ruedas. De esta forma, se evaluarán las trayectorias obtenidas mediante la construcción de modelos cinemáticos de movimiento y seremos capaces de describir el movimiento del WMR como un todo en función de su geometría y del comportamiento individual de cada una de sus ruedas.

2.2. Representación de la posición del robot

El WMR será modelado como un cuerpo rígido con ruedas. En el espacio tridimensional, un cuerpo rígido tiene seis DOF: tres traslacionales que describen la posición y tres rotacionales que describen la actitud. De aquí en adelante se puede utilizar el término *pose* para agrupar la posición y actitud. No serán tenidos en consideración los DOF adicionales producto de la flexibilidad o movilidad de los ejes y juntas. Sólo se tendrá en cuenta al chasis y las ruedas operando como un cuerpo rígido.

2.2.1. Marcos de referencia

La forma convencional para describir la pose de un cuerpo rígido es mediante marcos de referencia. De esta forma, la pose, desplazamientos o rotaciones del cuerpo rígido pueden ser expresados como la pose, desplazamientos o rotaciones de un marco de referencia “móvil” con respecto a un marco de referencia “fijo” (o inercial). Esto indica que el observador se encuentra en una posición estacionaria con respecto al marco de referencia fijo y no debe confundirse con la idea de que existe un marco de referencia **absoluto** [130]. Por ejemplo, un objeto a no tiene *una* velocidad. Tiene una velocidad con respecto a otro objeto del universo. Este otro objeto que elijamos para expresar la magnitud de interés será el marco de referencia. De esta forma, cuando escribimos \mathbf{u}_b^a , estamos representando la magnitud vectorial \mathbf{u} del objeto b **con respecto al marco de referencia a** . Con esto en cuenta, $\mathbf{v}_{bicicleta}^{auto}$, $\mathbf{v}_{bicicleta}^{tierra}$ y $\mathbf{v}_{bicicleta}^{persona}$ son tres de las tantas velocidades que pueden asociarse con una bicicleta en cualquier instante. Un marco de referencia puede verse como un conjunto de convenciones que permiten a un observador representar magnitudes físicas y consiste de dos elementos básicos: (1) la elección del origen, y (2) la elección de los vectores unitarios.

Para especificar la pose del robot en el espacio, se debe poder establecer una relación entre el marco de referencia fijo o inercial y el marco de referencia móvil del robot. Una vez definido el sistema coordenado del marco de referencia del cuerpo rígido, su pose puede ser totalmente descripta mediante la posición del origen de coordenadas y la orientación de cada uno de sus ejes, relativo al marco de referencia fijo elegido. La herramienta que nos permitirá describir la pose del WMR en el marco de referencia deseado es la *matriz de rotación*.

La posición \mathbf{p} del origen del marco de coordenadas m relativo con el marco de coordenadas f puede ser escrita como

$$\mathbf{p}_m^f = \begin{bmatrix} x_m^f \\ y_m^f \\ z_m^f \end{bmatrix}, \quad (2.1)$$

donde las coordenadas x_m^f , y_m^f y z_m^f son las proyecciones del origen del marco de referencia m en los ejes correspondientes del marco de referencia f . La actitud generalmente se representa mediante un conjunto de tres ángulos de Euler. Los ángulos de Euler son una elección popular debido a que son fáciles de entender y utilizar. Algunos conjuntos de ángulos de Euler hasta tienen nombre, por ejemplo: ángulo de dirección (*heading* o *yaw*), ángulo de elevación (*pitch*) y ángulo de alabeo (*roll*). Sin embargo, tienen sus desventajas: *i*) algunas funciones que derivan del uso de ángulos de Euler tienen

singularidades, y *ii*) son menos precisos que el cuaternión unitario cuando se los utiliza para integrar cambios incrementales en la actitud.

Estas deficiencias han llevado a que la comunidad utilice el cuaternión unitario para parametrizar la actitud de un cuerpo rígido. Las funciones relevantes de los cuaterniones unitarios no tienen singularidades y su representación es adecuada para integrar la velocidad angular con respecto al tiempo. Sin embargo, sus principales desventajas son: *i*) no tienen un significado físico intuitivo (a menudo se lo vuelve a transformar a ángulos de Euler para el análisis y visualización), y *ii*) debe tener norma unitaria para representar una rotación (esta restricción puede ser un problema cuando se la intenta incluir en algunos *solvers* de optimización).

2.2.2. Vectores, proyecciones y matrices de cosenos directores

Para representar el espacio en el que se desplaza el vehículo se utilizarán espacios vectoriales Euclídeos con producto interno definido. En el espacio Euclídeo n -dimensional es posible construir un conjunto ortonormal de n vectores $\mathcal{R} = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_n\}$. De esta forma, cualquier vector arbitrario \mathbf{v} puede representarse en el espacio definido por el conjunto \mathcal{R} mediante

$$\mathbf{v} = v_1\mathbf{r}_1 + v_2\mathbf{r}_2 + \dots + v_n\mathbf{r}_n \quad (2.2)$$

donde v_1, v_2, \dots, v_n se denominan las coordenadas de \mathbf{v} .

En el caso del espacio vectorial tridimensional, los vectores unitarios \mathbf{r}_1 , \mathbf{r}_2 y \mathbf{r}_3 pueden ser representados gráficamente por un conjunto de tres ejes ortogonales (supondremos que estos ejes siguen la regla de la mano derecha) y las coordenadas v_1, v_2, v_3 del vector unitario \mathbf{v} son las proyecciones de \mathbf{v} sobre $\{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3\}$. Cada una de estas proyecciones puede obtenerse utilizando el producto interno entre vectores

$$v_i = \mathbf{v} \cdot \mathbf{r}_i, \quad i = 1, 2, \dots, n. \quad (2.3)$$

El coseno del ángulo entre los vectores \mathbf{u} y \mathbf{v} , denotado como $\cos(\mathbf{u}, \mathbf{v})$ es determinado por la ley de los cosenos por la siguiente ecuación

$$\cos(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{|\mathbf{u}||\mathbf{v}|}, \quad (2.4)$$

como \mathbf{v} es un vector unitario

$$v_i = \mathbf{v} \cdot \mathbf{r}_i = \cos(\mathbf{v}, \mathbf{r}_i), \quad i = 1, 2, \dots, n \quad (2.5)$$

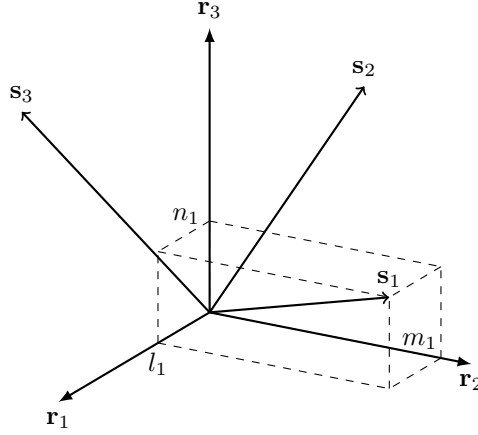


Figura 2.1: Proyección de \mathbf{s}_1 en \mathcal{R} .

por lo que cada una de las coordenadas de \mathbf{v} expresadas en \mathcal{R} pueden ser representadas mediante el producto interno o por el coseno del ángulo entre \mathbf{v} y cada uno de los vectores del conjunto \mathcal{R} .

Consideremos dos conjuntos de vectores ortonormales: $\mathcal{R} = \{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3\}$ y $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3\}$. Si tomamos al conjunto \mathcal{R} como nuestro marco de referencia y definimos con l_1 , m_1 y n_1 a las proyecciones del vector \mathbf{s}_1 sobre $\{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3\}$ (ver Figura 2.1), de la Ecuación (2.4), podemos ver que

$$[l_1, m_1, n_1] = [\cos(\mathbf{s}_1, \mathbf{r}_1), \cos(\mathbf{s}_1, \mathbf{r}_2), \cos(\mathbf{s}_1, \mathbf{r}_3)]. \quad (2.6)$$

Por este motivo, las coordenadas (l_1, m_1, n_1) se denominan los *cosenos directores* de \mathbf{s}_1 . Como hemos visto, el vector \mathbf{s}_1 puede ser expresado en el marco de referencia \mathcal{R} como

$$\mathbf{s}_1 = l_1\mathbf{r}_1 + m_1\mathbf{r}_2 + n_1\mathbf{r}_3. \quad (2.7)$$

Si procedemos de la misma manera las proyecciones de \mathbf{s}_2 y \mathbf{s}_3 están dadas por

$$\begin{aligned} \mathbf{s}_2 &= l_2\mathbf{r}_1 + m_2\mathbf{r}_2 + n_2\mathbf{r}_3, \\ \mathbf{s}_3 &= l_3\mathbf{r}_1 + m_3\mathbf{r}_2 + n_3\mathbf{r}_3. \end{aligned} \quad (2.8)$$

Este conjunto de ecuaciones que representa la proyección de $\{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3\}$ en el marco de referencia \mathcal{R} puede ser escrito de forma matricial como

$$[\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3] = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3] \begin{bmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \\ n_1 & n_2 & n_3 \end{bmatrix}, \quad (2.9)$$

al transponer esta ecuación obtenemos

$$\begin{bmatrix} \mathbf{s}_1^T \\ \mathbf{s}_2^T \\ \mathbf{s}_3^T \end{bmatrix} = \begin{bmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{bmatrix} \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix}. \quad (2.10)$$

Si llamamos \mathbf{C} a la matriz del lado derecho de la ecuación anterior,

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \mathbf{c}_3 \end{bmatrix} = \begin{bmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{bmatrix}, \quad (2.11)$$

como cada uno de los elementos de esta matriz es un término que involucra un coseno (Ecuación (2.6)), es denominada matriz de cosenos directores (DCM, Direction Cosine Matrix). La DCM es de gran importancia para determinar la orientación de cuerpos rígidos, ya que define la rotación de un marco relativo a otro. También puede ser utilizada para determinar las coordenadas en \mathcal{S} de un vector arbitrario en \mathcal{R} (o viceversa).

La DCM es una matriz ortogonal

Una matriz unitaria es aquella matriz cuya inversa es su traspuesta. A partir de (2.7)- (2.8) y utilizando la propiedad de ortonormalidad del conjunto \mathcal{R} , podemos demostrar que

$$\begin{aligned} \mathbf{s}_i \cdot \mathbf{s}_i &= l_i^2 + m_i^2 + n_i^2, \quad i = 1, 2, 3 \\ \mathbf{s}_i \cdot \mathbf{s}_j &= l_i l_j + m_i m_j + n_i n_j, \quad i = 1, 2, 3; i \neq j \end{aligned} \quad (2.12)$$

De esta forma, el producto $\mathbf{C}\mathbf{C}^T$ estará dado por

$$\begin{aligned} \mathbf{C}\mathbf{C}^T &= \begin{bmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{bmatrix} \begin{bmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \\ n_1 & n_2 & n_3 \end{bmatrix} \\ &= \begin{bmatrix} l_1^2 + m_1^2 + n_1^2 & l_1 l_2 + m_1 m_2 + n_1 n_2 & l_1 l_3 + m_1 m_3 + n_1 n_3 \\ l_2 l_1 + m_2 m_1 + n_2 n_1 & l_2^2 + m_2^2 + n_2^2 & l_2 l_3 + m_2 m_3 + n_2 n_3 \\ l_3 l_1 + m_3 m_1 + n_3 n_1 & l_3 l_2 + m_3 m_2 + n_3 n_2 & l_3^2 + m_3^2 + n_3^2 \end{bmatrix} \end{aligned}$$

utilizando (2.12), obtenemos $\mathbf{C}\mathbf{C}^T = \mathbf{I}$, con lo que se demuestra que la traspuesta de \mathbf{C} equivale a su inversa.

La DCM es una matriz de transformación

Una matriz de transformación es aquella matriz que transforma las coordenadas de un vector de un marco a otro [9]. La matriz que realiza esta transformación es la DCM. Dado que las coordenadas de un vector dependen del marco en el que se lo represente, un vector arbitrario \mathbf{v} puede ser representado en el marco \mathcal{R} como

$$\mathbf{v} = (\mathbf{v} \cdot \mathbf{r}_1)\mathbf{r}_1 + (\mathbf{v} \cdot \mathbf{r}_2)\mathbf{r}_2 + (\mathbf{v} \cdot \mathbf{r}_3)\mathbf{r}_3 \quad (2.13)$$

en notación matricial, podemos expresar la misma ecuación de la siguiente forma

$$\mathbf{v} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{bmatrix} \begin{bmatrix} v_{r1} \\ v_{r2} \\ v_{r3} \end{bmatrix}, \quad (2.14)$$

donde

$$v_{ri} = \mathbf{v} \cdot \mathbf{r}_i = \mathbf{r}_i^T \mathbf{v}, \quad i = 1, 2, 3. \quad (2.15)$$

De la misma manera, en el marco \mathcal{S} , el vector \mathbf{v} se expresa

$$\mathbf{v} = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \mathbf{s}_3 \end{bmatrix} \begin{bmatrix} v_{s1} \\ v_{s2} \\ v_{s3} \end{bmatrix}, \quad (2.16)$$

donde

$$v_{si} = \mathbf{v} \cdot \mathbf{s}_i = \mathbf{s}_i^T \mathbf{v}, \quad i = 1, 2, 3. \quad (2.17)$$

Para distinguir las proyecciones de \mathbf{v} en el marco \mathcal{R} y \mathcal{S} , sus respectivas coordenadas serán denotadas como

$$\mathbf{v}^r = \begin{bmatrix} v_{r1} \\ v_{r2} \\ v_{r3} \end{bmatrix} \quad y \quad \mathbf{v}^s = \begin{bmatrix} v_{s1} \\ v_{s2} \\ v_{s3} \end{bmatrix} \quad (2.18)$$

de manera que \mathbf{v}^r y \mathbf{v}^s se relacionan a través de

$$\begin{aligned} \mathbf{v}^s &= \begin{bmatrix} v_{s1} \\ v_{s2} \\ v_{s3} \end{bmatrix} = \begin{bmatrix} \mathbf{s}_1^T \mathbf{v} \\ \mathbf{s}_2^T \mathbf{v} \\ \mathbf{s}_3^T \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{s}_1^T \\ \mathbf{s}_2^T \\ \mathbf{s}_3^T \end{bmatrix} \mathbf{v} = \begin{bmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{bmatrix} \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix} \mathbf{v} \\ &= \begin{bmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{bmatrix} \begin{bmatrix} \mathbf{r}_1^T \mathbf{v} \\ \mathbf{r}_2^T \mathbf{v} \\ \mathbf{r}_3^T \mathbf{v} \end{bmatrix} = \begin{bmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{bmatrix} \mathbf{v}^r \end{aligned} \quad (2.19)$$

Este resultado puede resumirse como

$$\mathbf{v}^s = \begin{bmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{bmatrix} \mathbf{v}^r = \mathbf{C}_r^s \mathbf{v}^r, \quad (2.20)$$

donde se utiliza la notación \mathbf{C}_r^s para describir la DCM que transforma las coordenadas del vector del marco \mathcal{R} al marco \mathcal{S} . Si hubiéramos cambiado los roles de \mathbf{r} y \mathbf{s} , tendríamos el resultado

$$\mathbf{v}^r = \mathbf{C}_s^r \mathbf{v}^s. \quad (2.21)$$

Si utilizamos (2.20), obtenemos

$$\mathbf{v}^r = \mathbf{C}_s^r \mathbf{C}_r^s \mathbf{v}^r, \quad (2.22)$$

como \mathbf{v} es un vector arbitrario, eso implica que $\mathbf{C}_s^r \mathbf{C}_r^s = \mathbf{I}$. Y como la DCM es una matriz unitaria, $\mathbf{C}_s^{rT} = \mathbf{C}_r^s$.

2.2.3. Matrices de rotación

Las matrices de rotación son la herramienta que nos permite representar rotaciones arbitrarias. De esta forma, la actitud de un cuerpo rígido relativo a un marco de referencia, será expresada utilizando rotaciones de coordenadas. Una vez que el marco de referencia inercial esté definido, veremos que sus ejes determinan las columnas de la matriz de rotación que relaciona el marco inercial con el del vehículo y nos permitirá expresar la orientación en el marco de elección. En esta sección se va a describir como crear matrices de rotación y se mostrará su equivalencia con las matrices de transformación.

Para comenzar, veremos como describir la rotación de un vector alrededor de un eje arbitrario. Para ello, podemos aplicar el *teorema de rotación de Euler*: cualquier rotación o secuencia de rotaciones de un cuerpo rígido o sistema de coordenadas sobre un punto fijo es equivalente a una sola rotación sobre un eje fijo (o eje de Euler) que pasa por el punto fijo. La Figura 2.2 muestra un eje de rotación a lo largo del vector unitario \mathbf{u} , sobre el cual puede rotar cualquier vector. En este caso, el vector \mathbf{a} (representado como $\vec{m}\mathbf{a}$) intersecta a \mathbf{u} en el punto m . Como \mathbf{a} rota alrededor de \mathbf{u} , el punto a dibujará un círculo centrado en o . El vector $\vec{o}\mathbf{b}$ se construye de manera tal que se encuentre en el mismo plano que el círculo y que forme un ángulo recto con $\vec{o}\mathbf{a}$ y supongamos que \mathbf{a} realiza una rotación de un ángulo ϕ sobre \mathbf{u} para moverse a \mathbf{d} . Nuestro objetivo es describir \mathbf{d} como una función de \mathbf{a} , \mathbf{u} y ϕ .

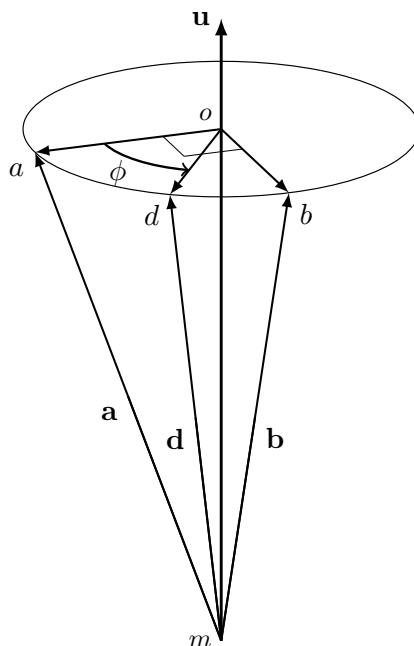


Figura 2.2: Rotación de un vector \mathbf{a} sobre el eje \mathbf{u} a una nueva posición \mathbf{d} .

Se puede ver que \mathbf{a} es la suma de dos vectores ortogonales: $\vec{m}o$, la proyección de \mathbf{a} sobre \mathbf{u} y $\vec{o}a$, la proyección de \mathbf{a} en el plano del círculo. De esta forma, utilizando la Ecuación (2.3) podemos escribir

$$\begin{aligned}
 \vec{m}o &= (\mathbf{u} \cdot \mathbf{a})\mathbf{u}, \\
 &= (\mathbf{u}^T \mathbf{a})\mathbf{u}, \\
 &= \mathbf{u}(\mathbf{u}^T \mathbf{a}), \\
 &= \mathbf{u}\mathbf{u}^T \mathbf{a},
 \end{aligned} \tag{2.23}$$

por lo que el vector $\vec{o}a$ puede obtenerse mediante

$$\begin{aligned}
 \vec{o}a &= \vec{m}a - \vec{m}o, \\
 &= \mathbf{a} - \mathbf{u}\mathbf{u}^T \mathbf{a}, \\
 &= (\mathbf{I} - \mathbf{u}\mathbf{u}^T)\mathbf{a}.
 \end{aligned} \tag{2.24}$$

Si tenemos en cuenta que el vector $\vec{o}b$ es ortogonal a los vectores \mathbf{u} y $\vec{o}a$, además de que $|\vec{o}b| = |\vec{o}a|$, entonces

$$\vec{o}b = \mathbf{u} \times \vec{o}a. \tag{2.25}$$

Si utilizamos el operador $[\mathbf{u}]^\times$ y (2.24), obtenemos

$$\begin{aligned}\vec{o}\vec{b} &= [\mathbf{u}]^\times \vec{o}\vec{a}, \\ &= [\mathbf{u}]^\times (\mathbf{I} - \mathbf{u}\mathbf{u}^T)\mathbf{a}, \\ &= [\mathbf{u}]^\times \mathbf{a},\end{aligned}\tag{2.26}$$

donde el operador $[\mathbf{u}]^\times$ esta definido como el equivalente matricial a la operación que realiza el producto cruz

$$[\mathbf{u}]^\times = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}.\tag{2.27}$$

El vector $\vec{o}\vec{d}$ expresado en término de su proyección sobre los vectores ortogonales $\vec{o}\vec{a}$ y $\vec{o}\vec{b}$ es

$$\vec{o}\vec{d} = \cos \phi \vec{o}\vec{a} + \sin \phi \vec{o}\vec{b},\tag{2.28}$$

de esta forma, el \mathbf{d} (o $\vec{m}\vec{d}$) es

$$\begin{aligned}\mathbf{d} &= \vec{m}\vec{o} + \vec{o}\vec{d}, \\ &= \vec{m}\vec{o} + \cos \phi \vec{o}\vec{a} + \sin \phi \vec{o}\vec{b}.\end{aligned}\tag{2.29}$$

Utilizando las ecuaciones (2.23), (2.24) y (2.26) para reemplazar $\vec{m}\vec{o}$, $\vec{o}\vec{a}$ y $\vec{o}\vec{b}$ en término de su forma matricial, se obtiene

$$\begin{aligned}\mathbf{d} &= \mathbf{u}\mathbf{u}^T \mathbf{a} + \cos \phi (\mathbf{I} - \mathbf{u}\mathbf{u}^T)\mathbf{a} + \sin \phi [\mathbf{u}]^\times \mathbf{a} \\ &= \left[\mathbf{u}\mathbf{u}^T + \cos \phi (\mathbf{I} - \mathbf{u}\mathbf{u}^T) + \sin \phi [\mathbf{u}]^\times \right] \mathbf{a} \\ &= \left[\cos \phi \mathbf{I} + (1 - \cos \phi)\mathbf{u}\mathbf{u}^T + \sin \phi [\mathbf{u}]^\times \right] \mathbf{a}.\end{aligned}\tag{2.30}$$

Si denominamos \mathbf{C} a la matriz que pre-multiplica al vector \mathbf{a} , tenemos

$$\mathbf{C} = \cos \phi \mathbf{I} + (1 - \cos \phi)\mathbf{u}\mathbf{u}^T + \sin \phi [\mathbf{u}]^\times,\tag{2.31}$$

por lo que \mathbf{C} es la matriz que rota el vector \mathbf{a} alrededor del vector \mathbf{u} un ángulo ϕ a su nueva posición \mathbf{d} . La ecuación (2.31) es también conocida como la *fórmula de rotación* o *fórmula de Rodríguez* [41].

A pesar de que las matrices de rotación y de transformación comparten las mismas propiedades, no son lo mismo. La DCM de (2.31) corresponde a rotar un vector un ángulo ϕ mientras los ejes coordenados permanecen fijos. ¿Qué sucedería si dejamos el vector fijo y rotamos los ejes coordenados? Podemos

notar que si se hubiese dejado a \mathbf{v} fijo y rotado el marco de referencia un ángulo de $-\phi$, \mathbf{v} tendría la misma posición relativa. De esta forma, si se desea obtener la DCM que rota el marco de referencia un ángulo ϕ , podemos utilizar la DCM que equivale a rotar el vector \mathbf{u} un ángulo $-\phi$ [9] y de esta manera (2.31) cambia a

$$\mathbf{C} = \cos(-\phi) \mathbf{I} + (1 - \cos(-\phi))\mathbf{u}\mathbf{u}^T + \sin(-\phi) [\mathbf{u}]^\times, \quad (2.32)$$

o lo que equivale

$$\mathbf{C} = \cos \phi \mathbf{I} + (1 - \cos \phi)\mathbf{u}\mathbf{u}^T - \sin \phi [\mathbf{u}]^\times. \quad (2.33)$$

Denominaremos rotaciones simples a aquellas que rotan un *marco* sobre un eje. En este caso, analizaremos aquellas que se realizan sobre el eje x , el y o el z . Se utilizará (2.33) para construir la DCM correspondiente

- Eje x : la DCM que rota un ángulo ϕ alrededor de $\mathbf{u} = [1 \ 0 \ 0]^T$ es

$$\mathbf{C}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \quad (2.34)$$

- Eje y : la DCM que rota un ángulo θ alrededor de $\mathbf{u} = [0 \ 1 \ 0]^T$ es

$$\mathbf{C}_y(\theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.35)$$

- Eje z : la DCM que rota un ángulo ψ alrededor de $\mathbf{u} = [0 \ 0 \ 1]^T$ es

$$\mathbf{C}_z(\psi) = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.36)$$

Ángulos de Euler y cuaterniones

Al utilizar ángulos de Euler, la DCM se obtiene a partir de una secuencia determinada de rotaciones simples de cada uno de los ejes. De esta forma, la ubicación del cada una de las rotaciones sucesivas depende de la rotación por la cual fue precedido. Esto implica que no hay una secuencia de ejes determinada, x - y - z no es lo mismo que una y - z - x , por lo que es necesario

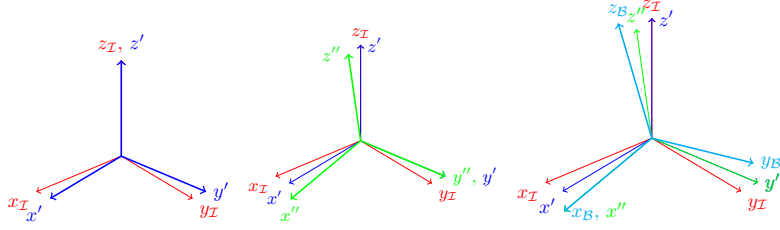


Figura 2.3: Secuencia de rotaciones correspondientes a cada eje coordenado.

dejar explícita la secuencia a utilizar. En este trabajo utilizaremos la secuencia $z-y-x$, donde primero se rota un ángulo ψ sobre el eje z , luego un ángulo θ sobre el eje y' para finalmente rotar un ángulo ϕ sobre el eje x'' (ver Figura 2.3).

Teniendo en cuenta que el producto de matrices de rotación es otra matriz de rotación, podemos obtener la matriz de rotación que rota $(z - y - x)$ mediante

$$\mathbf{C}_{zyx} = \mathbf{C}_x(\phi)\mathbf{C}_y(\theta)\mathbf{C}_z(\psi) \quad (2.37)$$

al realizar las operaciones, obtenemos

$$\mathbf{C}_{zyx} = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix}$$

Como hemos visto en la Sección 2.2.3, el teorema de rotación de Euler y la ecuación (2.31) nos permiten representar cualquier rotación en 3D como una combinación de un vector \mathbf{u} y un escalar ϕ , donde \mathbf{u} (generalmente unitario) representa el eje fijo y ϕ el ángulo de rotación. A continuación estudiará la forma de utilizar esta representación utilizando cuaterniones para rotar cualquier vector un ángulo arbitrario ϕ alrededor del eje \mathbf{u} .

Un cuaternión $\tilde{\mathbf{q}}$ puede ser representado como un vector de cuatro dimensiones

$$\tilde{\mathbf{q}} = [q_0 \quad q_1 \quad q_2 \quad q_3]^T \quad (2.38)$$

o de forma equivalente, por un escalar y un vector de tres dimensiones

$$\tilde{\mathbf{q}} = \begin{bmatrix} q_0 \\ \mathbf{q} \end{bmatrix} \quad (2.39)$$

con $\mathbf{q} = [q_1 \quad q_2 \quad q_3]$. Con esta notación, podemos representar el escalar a y

el vector \mathbf{b} como cuaterniones haciendo

$$\tilde{\mathbf{a}} = \begin{bmatrix} a \\ \mathbf{0} \end{bmatrix}, \tilde{\mathbf{b}} = \begin{bmatrix} 0 \\ \mathbf{b} \end{bmatrix}. \quad (2.40)$$

El adjunto $\tilde{\mathbf{q}}^*$, la norma $\|\tilde{\mathbf{q}}\|$ y la inversa $\tilde{\mathbf{q}}^{-1}$ del cuaternión $\tilde{\mathbf{q}}$ se definen como

$$\tilde{\mathbf{q}}^* = \begin{bmatrix} q_0 \\ -\mathbf{q} \end{bmatrix}, \quad (2.41)$$

$$\|\tilde{\mathbf{q}}\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}, \quad (2.42)$$

$$\tilde{\mathbf{q}}^{-1} = \frac{\tilde{\mathbf{q}}^*}{\|\tilde{\mathbf{q}}\|}, \quad (2.43)$$

si el cuaternión $\tilde{\mathbf{q}}$ tiene norma unitaria, su inversa es igual a su adjunto. El producto entre dos cuaterniones $\tilde{\mathbf{q}}$ y $\tilde{\mathbf{p}}$ se define como

$$\tilde{\mathbf{q}} \cdot \tilde{\mathbf{p}} = \begin{bmatrix} q_0 p_0 - \mathbf{q} \cdot \mathbf{p} \\ q_0 \mathbf{p} + p_0 \mathbf{q} \pm \mathbf{q} \times \mathbf{p} \end{bmatrix}. \quad (2.44)$$

El signo \pm en la definición del producto entre cuaterniones indica la influencia de un sistema de coordenadas que siga la regla de la mano derecha o izquierda, respectivamente.

Rotación con cuaterniones

Consideremos el vector unitario \mathbf{u} y el ángulo de giro ϕ . Si tomamos el vector de tres dimensiones \mathbf{v} y el cuaternión unitario construido como $\tilde{\mathbf{q}} = [\cos(\frac{\phi}{2}) \quad \sin(\frac{\phi}{2})\mathbf{u}]$, entonces

$$\tilde{\mathbf{q}}^* \tilde{\mathbf{v}} \tilde{\mathbf{q}} = \begin{bmatrix} 0 \\ \mathbf{C}\mathbf{v} \end{bmatrix} \quad (2.45)$$

donde \mathbf{C} es la matriz que realiza la rotación de un ángulo ϕ sobre el vector \mathbf{u} . Si realizamos el producto de cuaterniones

$$\tilde{\mathbf{q}}^* \tilde{\mathbf{v}} = \begin{bmatrix} q_0 \\ -\mathbf{q} \end{bmatrix} \begin{bmatrix} 0 \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{q} \cdot \mathbf{v} \\ q_0 \mathbf{v} - \mathbf{q} \times \mathbf{v} \end{bmatrix}$$

si posmultiplicamos por $\tilde{\mathbf{q}}$

$$\tilde{\mathbf{q}}^* \tilde{\mathbf{v}} \tilde{\mathbf{q}} = \begin{bmatrix} \mathbf{q} \cdot \mathbf{v} \\ q_0 \mathbf{v} - \mathbf{q} \times \mathbf{v} \end{bmatrix} \begin{bmatrix} q_0 \\ \mathbf{q} \end{bmatrix}, \quad (2.46)$$

expandiendo este producto, se obtiene

$$\tilde{\mathbf{q}}^* \tilde{\mathbf{v}} \tilde{\mathbf{q}} = \begin{bmatrix} q_0(\mathbf{q} \cdot \mathbf{v}) - q_0(\mathbf{q} \cdot \mathbf{v}) + \mathbf{q} \cdot (\mathbf{q} \times \mathbf{v}) \\ (\mathbf{q} \cdot \mathbf{v})\mathbf{q} + q_0(q_0\mathbf{v} - \mathbf{q} \times \mathbf{v}) + (q_0\mathbf{v} - \mathbf{q} \times \mathbf{v}) \times \mathbf{q} \end{bmatrix}. \quad (2.47)$$

Como

$$\begin{aligned} \mathbf{q} \cdot (\mathbf{q} \times \mathbf{v}) &= 0, \\ \mathbf{v} \times \mathbf{q} &= -\mathbf{q} \times \mathbf{v}, \\ (\mathbf{q} \times \mathbf{v}) \times \mathbf{q} &= (\mathbf{q} \cdot \mathbf{q})\mathbf{v} - (\mathbf{v} \cdot \mathbf{q})\mathbf{q}, \end{aligned}$$

entonces

$$\tilde{\mathbf{q}}^* \tilde{\mathbf{v}} \tilde{\mathbf{q}} = \begin{bmatrix} 0 \\ (\mathbf{q} \cdot \mathbf{v})\mathbf{q} + q_0^2\mathbf{v} - 2q_0\mathbf{q} \times \mathbf{v} - (\mathbf{q} \cdot \mathbf{q})\mathbf{v} + (\mathbf{v} \cdot \mathbf{q})\mathbf{q} \end{bmatrix}. \quad (2.48)$$

Si tomamos solo la parte vectorial y simplificamos

$$\begin{aligned} \mathbf{v} &= (\mathbf{q} \cdot \mathbf{v})\mathbf{q} + q_0^2\mathbf{v} - 2q_0\mathbf{q} \times \mathbf{v} - (\mathbf{q} \cdot \mathbf{q})\mathbf{v} + (\mathbf{v} \cdot \mathbf{q})\mathbf{q}, \\ &= (q_0^2 - (\mathbf{q} \cdot \mathbf{q}))\mathbf{v} + 2(\mathbf{q} \cdot \mathbf{v})\mathbf{q} - 2q_0\mathbf{q} \times \mathbf{v}. \end{aligned}$$

Si escribimos esta ecuación de forma matricial, se convierte en

$$\mathbf{v} = (q_0^2 - \mathbf{q}^T \mathbf{q})\mathbf{v} + 2\mathbf{q}\mathbf{q}^T \mathbf{v} - 2q_0[\mathbf{q}]^\times \mathbf{v} = \mathbf{C}\mathbf{v}, \quad (2.49)$$

donde $\mathbf{C} = (q_0^2 - \mathbf{q}^T \mathbf{q}) + 2\mathbf{q}\mathbf{q}^T - 2q_0[\mathbf{q}]^\times$. Por otro lado, si utilizamos los elementos de $\tilde{\mathbf{q}} = [\cos(\frac{\phi}{2}) \quad \sin(\frac{\phi}{2})\mathbf{u}]$ junto con algunas identidades trigonométricas, podemos ver

$$\begin{aligned} \cos \phi &= \cos^2\left(\frac{\phi}{2}\right) - \sin^2\left(\frac{\phi}{2}\right), \\ \sin \phi &= 2 \sin\left(\frac{\phi}{2}\right) \cos\left(\frac{\phi}{2}\right). \end{aligned}$$

Si sustituimos estos términos en la Ecuación (2.33)

$$\mathbf{C} = \left(\cos^2\left(\frac{\phi}{2}\right) - \sin^2\left(\frac{\phi}{2}\right) \right) \mathbf{I} + 2 \sin^2\left(\frac{\phi}{2}\right) \mathbf{u}\mathbf{u}^T - 2 \sin\left(\frac{\phi}{2}\right) \cos\left(\frac{\phi}{2}\right) [\mathbf{u}]^\times,$$

Si recordamos que $q_0 = \cos(\frac{\phi}{2})$ y $\mathbf{q} = \sin(\frac{\phi}{2})\mathbf{u}$, tenemos

$$\mathbf{q}^T \mathbf{q} = \sin^2\left(\frac{\phi}{2}\right) \mathbf{u}^T \mathbf{u} = \sin^2\left(\frac{\phi}{2}\right) [\mathbf{q}]^\times = \sin\left(\frac{\phi}{2}\right) [\mathbf{u}]^\times,$$

reemplazando

$$C = (q_0^2 - \mathbf{q}^T \mathbf{q})\mathbf{I} + 2\mathbf{q}\mathbf{q}^T - 2q_0[\mathbf{q}]^\times. \quad (2.50)$$

De esta forma, vemos que el producto $\tilde{\mathbf{q}}^* \tilde{\mathbf{v}} \tilde{\mathbf{q}}$ con $\tilde{\mathbf{q}} = [\cos(\frac{\phi}{2}) \quad \sin(\frac{\phi}{2})\mathbf{u}]$ equivale a multiplicar el vector \mathbf{v} por la matriz \mathbf{C} que rota un ángulo ϕ sobre el vector \mathbf{u} .

Cabe destacar que el producto $\tilde{\mathbf{q}}^* \tilde{\mathbf{v}} \tilde{\mathbf{q}}$ puede ser interpretado como una rotación del marco de coordenadas con respecto al espacio de puntos (fijo), en otras palabras, el vector \mathbf{v} no cambia. Asimismo, el producto $\tilde{\mathbf{q}} \tilde{\mathbf{v}} \tilde{\mathbf{q}}^*$ puede ser interpretado como una rotación del vector \mathbf{v} con respecto al marco de coordenadas (fijo). De modo que, el uso de cuaterniones nos brinda dos alternativas para transformar un vector:

1. Mediante el producto $\tilde{\mathbf{q}}^* \tilde{\mathbf{v}} \tilde{\mathbf{q}}$,
2. Convirtiendo el cuaternión a DCM mediante (2.50) y luego realizar la transformación.

2.2.4. Dinámica de las matrices de transformación

Las matrices de transformación (y rotación) nos permiten transformar un vector de un marco a otro. Hasta ahora, tanto el vector que determina el eje fijo de rotación y la DCM han sido considerado estáticos. ¿Qué sucedería si un marco cambia continuamente su dirección? Específicamente, consideraremos el caso en el que un marco rota relativo a otro a una velocidad angular $\boldsymbol{\omega}$.

Se debe determinar como la matriz de transformación varía con el tiempo; esto es, obtener su derivada con respecto al tiempo. A continuación analizaremos esta situación para los ángulos de Euler y el cuaternión unitario.

Ecuación diferencial de los ángulos de Euler

Las derivadas de los ángulos de Euler se obtienen como funciones de la velocidad angular con respecto a un marco de referencia. Para ello, se debe observar que los ángulos de Euler, y por consiguiente, sus derivadas, son calculados mediante una sucesión de rotaciones de distintos marcos. Si los convertimos a todos ellos al marco de referencia del cuerpo, obtenemos

$$\boldsymbol{\omega} = \mathbf{C}_{x''}(\phi) \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \mathbf{C}_{x''}(\phi)\mathbf{C}_{y'}(\theta) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathbf{C}_{x''}(\phi)\mathbf{C}_{y'}(\theta)\mathbf{C}_z(\psi) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}. \quad (2.51)$$

Al expandir y simplificar esta expresión, obtenemos

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \dot{\phi} - \dot{\psi} \sin(\theta) \\ \dot{\theta} \cos(\phi) + \dot{\psi} \sin(\phi) \cos(\theta) \\ -\dot{\theta} \sin(\phi) + \dot{\psi} \cos(\phi) \cos(\theta) \end{bmatrix}.$$

Si resolvemos este sistema para obtener las derivadas de los ángulos de Euler,

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}.$$

Las derivadas en (2.2.4) pueden tener problemas cuando θ se aproxime a 90 grados. Este fenómeno se conoce como *gimball lock* [123]. En el caso de los WMR podemos estar tranquilos que el ángulo de rotación θ alrededor del eje y no llegará a esos valores, sin embargo, es una particularidad que cabe ser destacada.

Ecuación diferencial del cuaternión

Consideremos el caso de dos marcos a y b inicialmente coincidentes, en el que el marco b rota relativo al marco fijo a . A tiempo t , la matriz de transformación de b a a estará determinada por

$$\mathbf{Q}_a^b(t) = \mathbf{Q}_a^{b(t)} \quad (2.52)$$

de esta forma, a tiempo $t + dt$

$$\mathbf{Q}_a^b(t + dt) = \mathbf{Q}_a^{b(t+dt)} = \mathbf{Q}_a^{b(t)} \mathbf{Q}_{b(t)}^{b(t+dt)} = \mathbf{Q}_a^b(t) \mathbf{Q}_{b(t)}^{b(t+dt)} \quad (2.53)$$

Supondremos que en el intervalo $[t, t + dt]$, el vector de velocidad angular del marco \mathbf{b} es

$$\boldsymbol{\omega} = w \mathbf{u} \quad (2.54)$$

donde w es la velocidad angular y \mathbf{u} es un vector unitario representado en el marco de referencia \mathbf{b} . Entonces, en el intervalo dt , el marco de referencia \mathbf{b} rotará un ángulo determinado por

$$\phi = w dt \quad (2.55)$$

Como hemos visto, el cuaternión que rota un ángulo ϕ sobre la dirección del vector \mathbf{u} se conforma mediante

$$\tilde{\mathbf{q}}_{b(t)}^{b(t+dt)} = \begin{bmatrix} \cos \frac{\phi}{2} \\ \sin \frac{\phi}{2} \mathbf{u} \end{bmatrix}. \quad (2.56)$$

Suponiendo que dt es infinitesimal, ϕ también lo es y podemos aplicar las aproximaciones para ángulos pequeños

$$\begin{aligned}\cos\left(\frac{\phi}{2}\right) &= \cos\left(\frac{w dt}{2}\right) \cong 1, \\ \sin\left(\frac{\phi}{2}\right) &= \sin\left(\frac{w dt}{2}\right) \cong \frac{w dt}{2}.\end{aligned}$$

Al realizar estas sustituciones, obtenemos

$$\tilde{\mathbf{q}}_{b(t)}^{b(t+dt)} = \begin{bmatrix} \cos\frac{\phi}{2} \\ \sin\frac{\phi}{2} \mathbf{u} \end{bmatrix} \cong \begin{bmatrix} 1 \\ \frac{w dt}{2} \mathbf{u} \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{1}{2}\boldsymbol{\omega} dt \end{bmatrix},$$

$$\tilde{\mathbf{q}}_a^b(t+dt) = \tilde{\mathbf{q}}_a^b(t) \begin{bmatrix} 1 \\ \frac{1}{2}\boldsymbol{\omega} dt \end{bmatrix} = \tilde{\mathbf{q}}_a^b(t) + \tilde{\mathbf{q}}_a^b(t) \begin{bmatrix} 0 \\ \frac{1}{2}\boldsymbol{\omega} dt \end{bmatrix},$$

lo que implica

$$\frac{\tilde{\mathbf{q}}_a^b(t+dt) - \tilde{\mathbf{q}}_a^b(t)}{dt} = \tilde{\mathbf{q}}_a^b(t) \begin{bmatrix} 0 \\ \frac{1}{2}\boldsymbol{\omega} dt \end{bmatrix},$$

si tomamos el límite por el lado izquierdo mientras $dt \rightarrow 0$, obtenemos

$$\frac{d}{dt}\tilde{\mathbf{q}}_a^b(t) = \frac{1}{2}\tilde{\mathbf{q}}_a^b(t) \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{ab}^b \end{bmatrix},$$

en forma matricial

$$\frac{d}{dt} \begin{bmatrix} q_0 \\ \mathbf{q} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\boldsymbol{\omega}^T \\ \boldsymbol{\omega} & [\boldsymbol{\omega}]^\times \end{bmatrix} \begin{bmatrix} q_0 \\ \mathbf{q} \end{bmatrix},$$

de esta forma, hemos logrado relacionar el cuaternión con el vector de velocidad angular $\boldsymbol{\omega}$ para obtener la ecuación diferencial que determina la actitud utilizando el cuaternión unitario $\tilde{\mathbf{q}}$.

2.3. Cinemática de los robots móviles

En el caso general, un robot móvil puede estar articulado de varias maneras, puede tener que desplazarse sobre terreno arbitrario y debido a ésto, en ocasiones una o más de sus ruedas puede no hacer contacto sobre la superficie del terreno. Si suponemos que el contacto de cada rueda con el terreno está asegurado –ya sea por geometría o por el uso de suspensiones–, hay dos dificultades principales asociadas con el modelado cinemático de robots

móviles: la aparición de condiciones no lineales y de más restricciones que los grados de libertad. Las condiciones no lineales aparecen al controlar la dirección, debido a que es necesario utilizar funciones trigonométricas para relacionar el ángulo de giro con las velocidades de las ruedas. Para el caso en el que el sistema tenga restricciones de más, puede suceder que tengamos un conjunto $m > n$ de mediciones de velocidad y/o ángulos de giro que hagan que la solución del problema de estimación sea inconsistente con la solución para los n grados de libertad disponibles para controlar el vehículo.

En esta sección se estudiarán las relaciones cinemáticas básicas entre:

- la velocidad lineal y angular de un marco de coordenadas determinado, ubicado en el cuerpo del robot móvil,
- la velocidad lineal de un punto arbitrario cuya posición se corresponde con una rueda.

Estas transformaciones serán formuladas utilizando álgebra vectorial, por lo que se obtendrán expresiones sencillas, aún para casos generalizados.

2.3.1. Teorema del transporte

La clave de la técnica a utilizar reside en un teorema básico de la física, usado comúnmente en áreas tales como la dinámica del cuerpo rígido o en navegación inercial. Conocido como el *Teorema del Transporte* o *ecuación de Coriolis*, la cual relaciona las mediciones físicas con el estado de movimiento de un observador. Siguiendo la notación definida, \vec{u}_a^b representa la magnitud vectorial u del marco de referencia a con respecto al marco de referencia b . Denominamos a uno de los observadores “fijo” y le asociamos el marco de referencia f . De la misma manera, asociamos el marco de referencia m con el observador “móvil”. Ambos marcos de referencia pueden encontrarse en estados de movimiento arbitrarios con respecto a alguna otra cosa, pero los nombres nos facilitarán la tarea de concentrarnos en su movimiento *relativo*. Para cualquier movimiento relativo de los observadores, la derivada con respecto al tiempo de cualquier vector \vec{u} desde el punto de vista del observador fijo está relacionada con la derivada del mismo vector, pero desde el punto de vista del observador móvil por

$$\left. \frac{d}{dt} \right|_f \vec{u} = \left. \frac{d}{dt} \right|_m \vec{u} + \vec{\omega}_m^f \times \vec{u}. \quad (2.57)$$

2.3.2. Transformación de velocidades

La posición relativa entre los marcos de referencia f y m está dada por el vector \vec{r}_m^f (ver Figura 2.4). Si unimos un marco de referencia o a un objeto y suponemos que el observador móvil mide su posición \vec{r}_o^m y velocidad \vec{v}_o^m , nos preguntamos: ¿qué mediciones del objeto determinado por o obtendrá el observador fijo desde el marco de referencia f ? Los vectores posición entre el marco f , m y o se pueden obtener a partir de la suma vectorial:

$$\vec{r}_o^f = \vec{r}_m^f + \vec{r}_o^m. \quad (2.58)$$

La derivada del vector posición, calculada en el marco de referencia f está dada por

$$\left. \frac{d}{dt} \right|_f (\vec{r}_o^f) = \left. \frac{d}{dt} \right|_f (\vec{r}_m^f + \vec{r}_o^m) = \left. \frac{d}{dt} \right|_f (\vec{r}_m^f) + \left. \frac{d}{dt} \right|_f (\vec{r}_o^m). \quad (2.59)$$

Ahora podemos aplicar la ecuación de Coriolis al último término de la derecha para obtener la velocidad aparente del objeto o entre dos marcos de referencia sujetos a movimientos arbitrarios

$$\vec{v}_o^f = \vec{v}_m^f + \vec{\omega}_m^f \times \vec{r}_o^m + \vec{v}_o^m. \quad (2.60)$$

Si procedemos de la misma forma, la expresión para las aceleraciones estarán dadas por

$$\vec{a}_o^f = \vec{a}_m^f + \vec{\alpha}_m^f \times \vec{r}_o^m + \vec{\omega}_m^f \times (\vec{\omega}_m^f \times \vec{r}_o^m) + 2\vec{\omega}_m^f \times \vec{v}_o^m + \vec{a}_o^m. \quad (2.61)$$

Los resultados de velocidad y aceleración son el caso general para movimientos relativos y la diferenciación ya ha sido realizada en el caso general. Dicho de otra manera, éstas ecuaciones pueden ser aplicadas a marcos de referencia encadenados que se encuentren en movimiento relativo, como por ejemplo, para describir el movimiento de un cuerpo rígido causado por la rotación de varias ruedas.

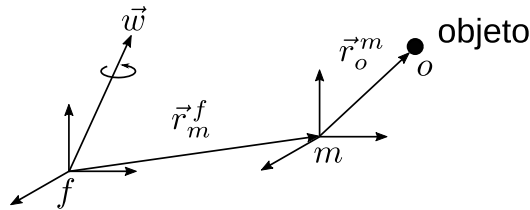


Figura 2.4: Marcos de referencia en movimiento: Dos observadores que se encuentran en movimiento relativo miden el movimiento del objeto.

2.3.3. Ecuación de una rueda

En el tratamiento generalizado de la ecuación para una rueda, vamos a considerar un modelo como el de la Figura 2.5 donde el punto sobre el cual se controlará la dirección (marco s) puede estar desplazado del punto de contacto de la rueda con el piso (marco c). Bajo tales hipótesis, podemos escribir el vector posición del punto de contacto de la rueda de la siguiente manera:

$$\vec{r}_c^w = \vec{r}_v^w + \vec{r}_s^v + \vec{r}_c^s \quad (2.62)$$

Si asociamos cualquier marco de referencia fijo en la tierra con el observador fijo y el marco de referencia del vehículo con el marco de referencia móvil, podemos utilizar la ecuación de transformación de velocidades para escribir la ecuación cinemática de la rueda. Al diferenciar el vector posición en el marco de referencia del mundo utilizando la ecuación de Coriolis y teniendo en cuenta que $\vec{v}_s^v = \vec{0}$, tenemos:

$$\vec{v}_c^w = \vec{v}_v^w + \vec{\omega}_v^w \times \vec{r}_s^v + \vec{\omega}_v^w \times \vec{r}_c^s + \vec{\omega}_c^v \times \vec{r}_c^s, \quad (2.63)$$

a esta ecuación la llamaremos la *ecuación de la rueda*. En el caso que no haya desplazamiento entre el eje de dirección y la rueda, el vector $\vec{r}_c^s = \vec{0}$ y los dos últimos términos desaparecen. Esta fórmula es válida en tres dimensiones y puede aplicarse para casos con articulaciones arbitrarias entre los marcos v y s , dado que solamente el vector \vec{r}_s^v es relevante. Dicho de otras palabras, este es el caso general.

El término **cinemática inversa** se refiere al problema, relevante al control, de calcular la velocidad correspondiente a cada rueda a partir de velocidades del vehículo. Esto se resuelve escribiendo una ecuación para cada

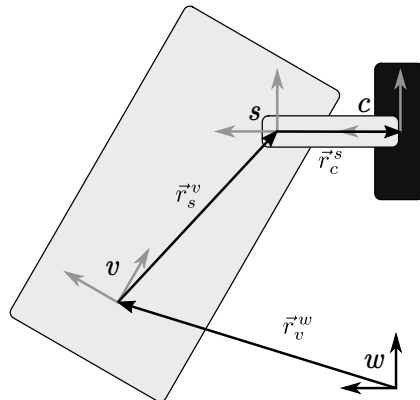


Figura 2.5: Marcos de referencia para la cinemática de una rueda.

rueda. Para ello, los vectores físicos \vec{u} deben ser expresados en un sistema de coordenadas particular. Para ello, se utilizará la notación ${}^c\bar{u}_a^b$ para denominar a la magnitud vectorial u del marco de referencia a con respecto al marco b , expresada en las coordenadas del marco c (y de forma implícita, ${}^b\bar{u}_a^b = \bar{u}_a^b$). De esta forma, si R_s^v es la matriz de rotación que convierte las coordenadas del marco s al marco v , es posible expresar la ecuación de cualquier rueda en el marco del vehículo, donde la mayoría de los vectores son conocidos:

$${}^v\bar{v}_c^w = {}^v\bar{v}_v^w + {}^v\bar{\omega}_v^w \times {}^v\bar{r}_s^v + {}^v\bar{\omega}_v^w \times R_s^{vs}\bar{r}_c^s + {}^v\bar{\omega}_c^v \times R_s^{vs}\bar{r}_c^s. \quad (2.64)$$

El término **cinemática directa** se refiere al problema, relevante a la estimación, de calcular la velocidad del vehículo a partir de las mediciones de las ruedas. La ecuación de la rueda puede ser escrita en forma matricial utilizando la matriz antisimétrica que realiza la operación equivalente al producto cruz de dos vectores mediante una multiplicación matricial dada por [2.27](#) ($a \times b = -b \times a = -[b]^\times a = ([b]^\times)^T a$). De esta forma, para cada rueda obtenemos

$${}^v\bar{v}_c^w = {}^v\bar{v}_v^w + ([{}^v\bar{r}_s^v]^\times)^T {}^v\bar{\omega}_v^w + ([{}^v\bar{r}_c^s]^\times)^T {}^v\bar{\omega}_v^w + ([{}^v\bar{r}_c^s]^\times)^T {}^v\bar{\omega}_c^v. \quad (2.65)$$

Para múltiples ruedas, cada una de estas ecuaciones se puede agrupar de forma matricial para obtener una ecuación de la forma:

$${}^v\bar{v}_c^w = H_v(\bar{\theta}) \begin{bmatrix} {}^v\bar{v}_v^w \\ {}^v\bar{\omega}_v^w \end{bmatrix} + H_\theta(\bar{\theta}) {}^v\bar{\omega}_c^w = H_v(\bar{\theta}) \bar{V} + H_\theta(\bar{\theta}) \dot{\bar{\theta}}, \quad (2.66)$$

donde ${}^v\bar{v}_c^w$ es la velocidad de cada rueda, \bar{V} son las velocidades lineales y angulares del vehículo con respecto al marco inercial. Cabe destacar que tanto ${}^v\bar{v}_c^w$ como \bar{V} están expresadas en el marco de referencia del vehículo. Los ángulos de dirección están expresados por $\bar{\theta}$ y puede incluir otras articulaciones o suspensiones en caso de ser necesario. El último término, $\dot{\bar{\theta}}$ es el incremento de la velocidad de la rueda debido a los cambios en las direcciones.

2.4. Tracción diferencial

Los WMRs pueden tener distintas un gran número de combinaciones posibles para sus ruedas y diseños cinemáticos. Cada tipo de configuración tiene sus ventajas y desventajas a ser consideradas con respecto al campo de aplicación del WMR. En esta sección se describirá la tracción diferencial.

La configuración de tracción diferencial es posiblemente una de las configuraciones más utilizada en robots móviles [\[6, 33\]](#), consiste en dos ruedas

principales montadas sobre un mismo eje de longitud L_y , cada una conectada a su propio motor, logrando proveer conjuntamente las funciones de tracción y de dirección (ver Figura 2.6).

El análisis del movimiento de un WMR de tracción diferencial es sencillo. Si denominamos v_l y v_r a la velocidad lineal de la rueda izquierda y derecha, respectivamente, podemos describir algunos casos de interés. Cuando $v_l = v_r$ el WMR se mueve en línea recta. Cuando $v_l = -v_r$, el WMR el robot rota sobre si mismo alrededor del punto medio del eje que une las dos ruedas. Esto hace que la configuración de tracción diferencial sea una opción atractiva cuando el WMR debe moverse en ambientes estrechos. Para otras configuraciones de v_l y v_r , el WMR no se mueve en línea recta, sino que describe trayectorias curvas o arcos, donde la velocidad lineal y angular del cuerpo dependerán de la velocidad de cada rueda. Por este motivo, un WMR de tracción diferencial es muy sensible a la velocidad relativa de las dos ruedas. Pequeños errores en la velocidad de cada rueda resulta en trayectorias diferentes, no solo en un WMR que se desplace más lento o más rápido. Generalmente, los WMR de tracción diferencial utilizan una rueda de soporte para estabilizar el vehículo, motivo por el cual son muy sensibles a variaciones en el plano sobre el cual el WMR se desplaza y limita su apli-



Figura 2.6: Lego NXT configurado como un robot móvil de tracción diferencial.

cación a ambientes de exteriores. Finalmente, se debe tener en cuenta que la estructura cinemática de la configuración de tracción diferencial prohíbe ciertos movimientos del WMR. Por ejemplo, no existe combinación de velocidades v_l y v_r que logren que el vehículo se mueva en dirección del eje que une las ruedas.

2.4.1. Tracción diferencial en 2D

Para el caso del modelado de un robot de tracción diferencial en 2D, se considera que el WMR se mueve sobre el plano x - y y que gira un ángulo ψ sobre el eje z . No se tiene en cuenta los ángulos de giro ϕ sobre el eje x y θ sobre el eje y . Gracias a estas simplificaciones, el modelo del WMR consta de dos ruedas unidas por un eje, tal como se aprecia en la Figura 2.7.

Llamaremos l y r a los marcos de referencia de la rueda izquierda y derecha, respectivamente. Sea v la velocidad lineal y ω la velocidad angular del marco de referencia b , ubicado en el cuerpo del robot. De esta forma, los vectores que describen la posición de cada una de las ruedas con respecto al cuerpo son:

$$\vec{r}_r^b = [0, -L_y]^T \qquad \vec{r}_l^b = [0, L_y]^T \qquad (2.67)$$

Si utilizamos la ecuación (2.58) para describir la velocidad de la rueda con respecto al marco de referencia inercial w , descrita en el marco de referencia del cuerpo b , obtenemos

$${}^b\vec{v}_r^w = {}^b\vec{v}_b^w + {}^b\vec{r}_r^b, \qquad (2.68)$$

al derivar utilizando la ecuación (2.60), obtenemos:

$$\begin{aligned} {}^b\vec{v}_r^w &= {}^b\vec{v}_b^w + {}^b\vec{\omega}_b^w \times {}^b\vec{r}_r^b + {}^b\vec{v}_r^b, \\ &= {}^b\vec{v}_b^w - [{}^b\vec{r}_r^b] \times {}^b\vec{\omega}_b^w + {}^b\vec{v}_r^b. \end{aligned} \qquad (2.69)$$

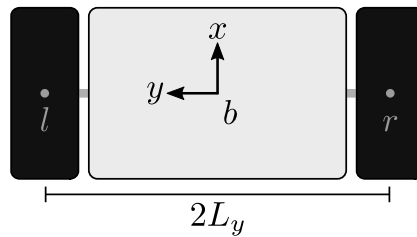


Figura 2.7: Tracción diferencial. En esta configuración hay dos ruedas montadas sobre un mismo eje.

Si aplicamos estas ecuaciones teniendo en cuenta el caso que estamos analizando, obtenemos:

$${}^b\vec{v}_r^w = \begin{bmatrix} v \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & -L_y \\ 0 & 0 & 0 \\ L_y & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix}, \quad (2.70)$$

de forma similar para el marco de referencia de la rueda izquierda, obtenemos

$${}^b\vec{v}_l^w = {}^b\vec{v}_b^w + {}^b\vec{\omega}_b^w \times {}^b\vec{r}_l^b + {}^b\vec{v}_l^b. \quad (2.71)$$

Concatenando las ecuaciones de cada rueda de forma matricial, se obtiene la expresión para la cinemática inversa del WMR de tracción diferencial:

$$\begin{bmatrix} {}^b\vec{v}_r^w \\ {}^b\vec{v}_l^w \end{bmatrix} = \begin{bmatrix} I & -[{}^b\vec{r}_r^b]^\times \\ I & -[{}^b\vec{r}_l^b]^\times \end{bmatrix} \begin{bmatrix} {}^b\vec{v}_b^w \\ {}^b\vec{\omega}_b^w \end{bmatrix}. \quad (2.72)$$

Para el caso particular de un robot de tracción diferencial en el plano, bajo la suposición de que las ruedas no presentan deslizamientos longitudinales ni laterales, obtenemos

$$\begin{bmatrix} {}^b\vec{v}_r^w \\ {}^b\vec{v}_l^w \end{bmatrix} = \begin{bmatrix} 1 & L_y \\ 1 & -L_y \end{bmatrix} \begin{bmatrix} {}^b\vec{v}_b^w \\ {}^b\vec{\omega}_b^w \end{bmatrix}. \quad (2.73)$$

Para obtener la cinemática inversa en las coordenadas del cuerpo, la ecuación (2.73) es sencilla de invertir, por lo tanto

$$\begin{bmatrix} {}^b\vec{v}_b^w \\ {}^b\vec{\omega}_b^w \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & \frac{1}{L_y} \\ 1 & -\frac{1}{L_y} \end{bmatrix} \begin{bmatrix} {}^b\vec{v}_r^w \\ {}^b\vec{v}_l^w \end{bmatrix}, \quad (2.74)$$

con lo que se obtiene una expresión que relaciona la velocidad lineal y angular del cuerpo a partir de la velocidad de cada una de las ruedas.

2.4.2. Tracción diferencial en 3D

Para el caso del modelado de un robot de tracción diferencial en 3D, debemos tener en cuenta la rueda de soporte (ver Figura 2.8), a fin de que el vehículo pueda girar con respecto al eje y .

Llamaremos l , r y c a los marcos de referencia de la rueda izquierda, derecha y de soporte, respectivamente. Sea v la velocidad lineal y ω la velocidad angular del marco de referencia b , ubicado en el cuerpo del robot. De esta

forma, los vectores que describen la posición de cada una de las ruedas con respecto al cuerpo son

$$\vec{r}_r^b = [L_x, -L_y, -L_z]^T, \quad (2.75)$$

$$\vec{r}_l^b = [L_x, L_y, -L_z]^T, \quad (2.76)$$

$$\vec{r}_c^b = [-C_x, 0, -C_z]^T. \quad (2.77)$$

De esta forma, la expresión para la cinemática inversa, que relaciona la velocidad lineal y angular del cuerpo con la velocidad de cada una de las tres ruedas es

$${}^b\vec{v}_r^w = \vec{v}_b^w - [{}^b\vec{r}_r^b]^\times \vec{\omega}_b^w, \quad (2.78)$$

$${}^b\vec{v}_l^w = \vec{v}_b^w - [{}^b\vec{r}_l^b]^\times \vec{\omega}_b^w, \quad (2.79)$$

$${}^b\vec{v}_c^w = \vec{v}_b^w - [{}^b\vec{r}_c^b]^\times \vec{\omega}_b^w. \quad (2.80)$$

Una vez obtenida la expresión para la cinemática inversa, se debe obtener las ecuaciones correspondientes a la cinemática directa, que relacionan la velocidad del vehículo a partir de las velocidades de las ruedas. Como la rueda de soporte, correspondiente al marco c no tiene motor, su velocidad es parte del sistema pero no es una variable a controlar, ya que depende de las velocidades de v_r y v_l . Sin embargo, esta rueda es necesaria para que el modelo tenga tres puntos de apoyo en el terreno, por lo que es necesario que v_c cumpla la condición

$$v_c \cdot \vec{n}_c = 0, \quad (2.81)$$

esto es, que las componentes de la velocidad de la rueda de soporte sean tangenciales a la normal en el punto de contacto (representado en coordenadas

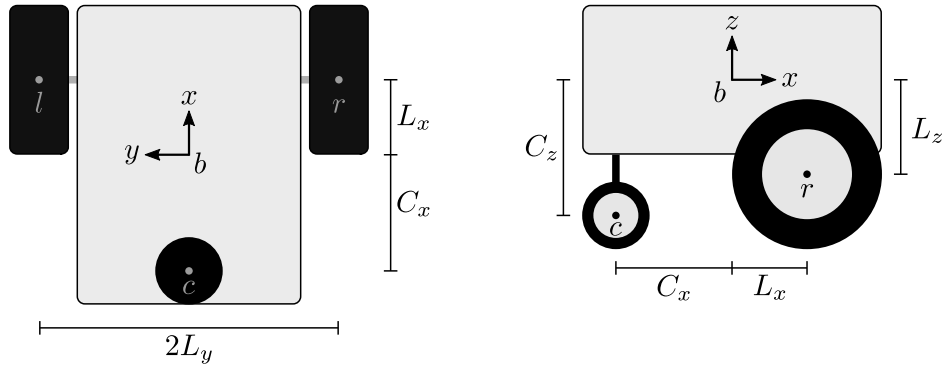


Figura 2.8: Tracción diferencial en 3D. Se agrega la rueda del soporte al modelo para tener en cuenta el ángulo de rotación θ sobre el eje y .

del vehículo). Agregando esta condición y reordenando de forma matricial obtenemos:

$$\begin{bmatrix} v_r \\ v_l \end{bmatrix} = \begin{bmatrix} I_{3 \times 3} & -S(r_r) & 0_{3 \times 3} \\ I_{3 \times 3} & -S(r_l) & 0_{3 \times 3} \\ I_{3 \times 3} & -S(r_c) & -I_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{2 \times 3} \cdot \vec{n}_c \end{bmatrix} \begin{bmatrix} v \\ \omega \\ v_c \end{bmatrix}. \quad (2.82)$$

Ahora resta obtener la proyección de v_r y v_l según el terreno. Para ello suponemos que no hay deslizamientos laterales, lo que equivale a decir que las componentes en la coordenada y de cada velocidad serán 0. El vector dirección de las velocidades deberá cumplir las siguientes condiciones

$$u \cdot \vec{n} = 0, \quad (2.83)$$

$$\|u\|_2^2 = 1. \quad (2.84)$$

La primer condición determina:

$$u_x n_x + 0 n_y + u_z n_z = 0 \implies u_z = -\frac{u_x n_x}{n_z}, \quad (2.85)$$

la segunda condición determina:

$$\begin{aligned} \|u\|_2^2 &= 1 \\ u_x^2 + u_y^2 + u_z^2 &= 1 \\ u_x^2 + \left(-\frac{u_x n_x}{n_z}\right)^2 &= 1 \\ u_x^2 + \frac{u_x^2 n_x^2}{n_z^2} &= 1 \\ u_x^2 \left(1 + \frac{n_x^2}{n_z^2}\right) &= 1 \\ u_x^2 \left(\frac{n_z^2 + n_x^2}{n_z^2}\right) &= 1 \\ u_x &= \sqrt{\frac{n_z^2}{n_z^2 + n_x^2}} \end{aligned} \quad (2.86)$$

Una vez obtenido el vector u , podemos multiplicarlo por la magnitud de cada una de las ruedas para obtener su proyección de acuerdo a la normal del terreno. Dicho sistema puede ser resuelto por mínimos cuadrados para obtener los valores de v , ω y v_c . Para profundizar en el tema el lector puede dirigirse a [9, 28, 46, 55].

2.5. Resultados

A continuación presentamos los resultados obtenidos a partir de simular el modelo cinemático para un robot de tracción diferencial en terreno plano y en presencia de una rampa inclinada, con estado inicial $x_0 = (0.1, 0.6, 0.1, 0, 0, 0)$ y aplicando la misma velocidad a cada una de las ruedas.

Ejemplo 1: Terreno plano

En este ejemplo, podemos observar que al aplicar la misma velocidad a cada rueda el robot se desplaza en línea recta hacia adelante. La Figura 2.9a muestra la evolución de la coordenada x , donde se puede observar que el robot avanza hasta llegar a los 40 m. Al observar las Figuras 2.9b y 2.9c podemos ver que las coordenadas y y z se mantienen constante con respecto a su estado inicial. Como es de esperarse, los ángulos de orientación también se mantienen constantes en todo momento (ver Figuras 2.9d, 2.9e y 2.9f).

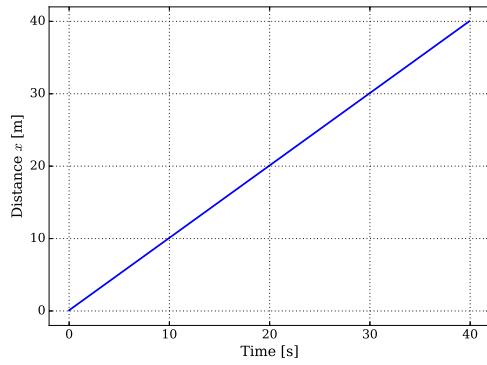
Ejemplo 2: Rampa inclinada

En este ejemplo, se puede observar que al aplicar la misma velocidad a cada una de las ruedas en presencia de una rampa inclinada el robot ya no se desplaza en línea recta, sino que su movimiento está determinado por la velocidad de cada rueda y por las características del terreno (ver Figura 2.10). Al observar la Figura 2.10b podemos ver que al ingresar a la rampa el vehículo se desplaza levemente hacia la derecha, para luego terminar su recorrido 70 cm a la izquierda de su coordenada inicial. De esta forma se puede deducir que la rueda izquierda estaba a una altura menor que la derecha, lo que también puede visualizarse en el ángulo de *roll* del vehículo (ver Figura 2.10d). De la misma forma, podemos ver que la coordenada z crece hasta llegar a los 26 cm, para luego mantenerse constante, por lo que se deduce que el vehículo estaba desplazándose en subida, lo cual también puede visualizarse en el ángulo de *pitch* (ver Figura 2.10e).

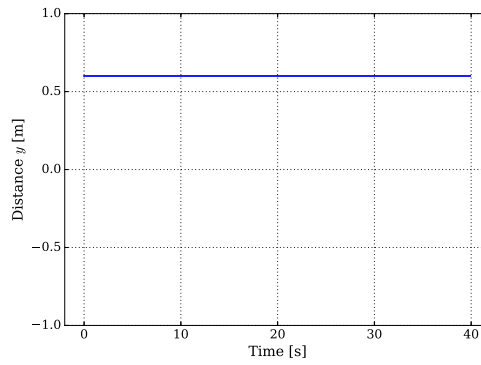
2.6. Conclusiones

En este capítulo se trató el modelado cinemático de robots móviles terrestres y se realizó una pequeña introducción su modelado 3-D. Las técnicas de modelado presentadas en este capítulo permiten generar de forma sistemática y generalizada modelos para robot móviles de las más diversas y complejas características. A partir de los resultados obtenidos mediante simulaciones, se

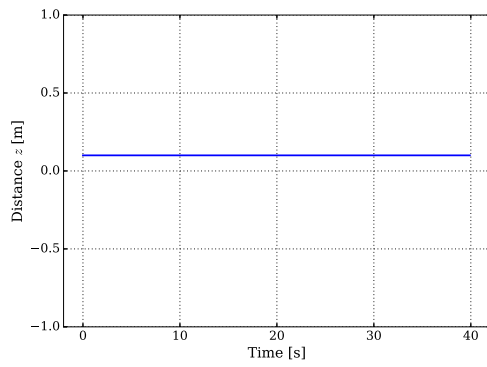
Conclusiones



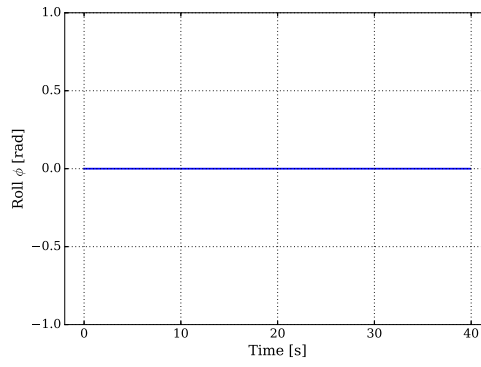
(a) Posición x .



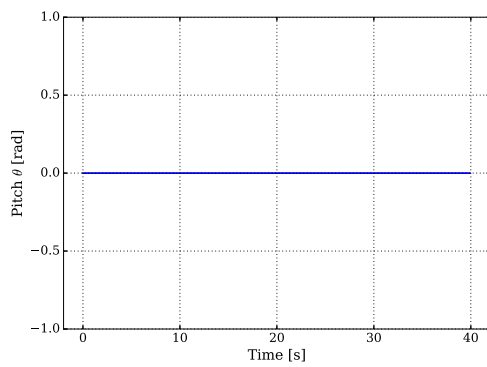
(b) Posición y .



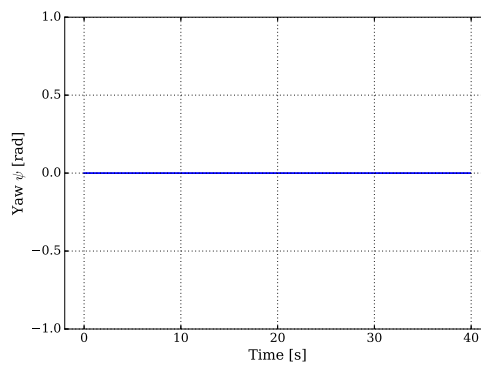
(c) Posición z .



(d) Ángulo de roll ϕ .

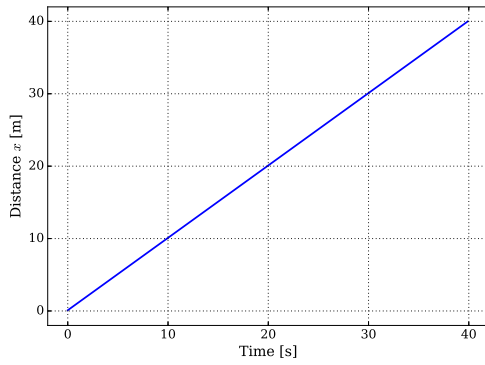


(e) Ángulo de pitch θ .

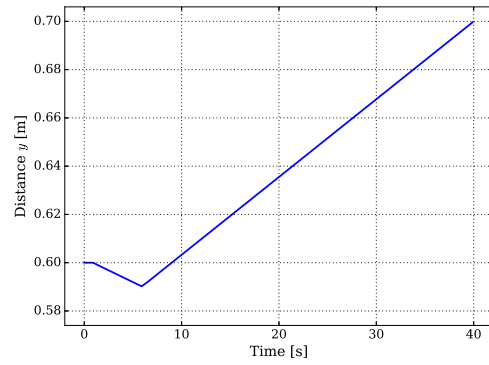


(f) Ángulo de yaw ψ .

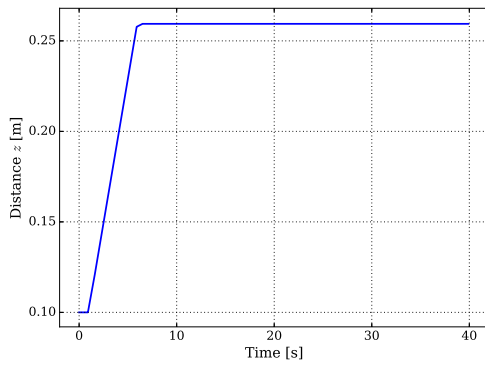
Figura 2.9: Simulación en terreno plano.



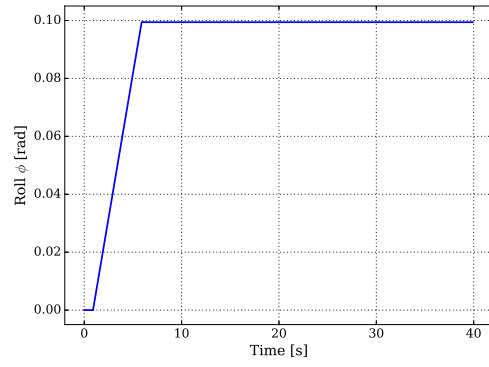
(a) Posición x .



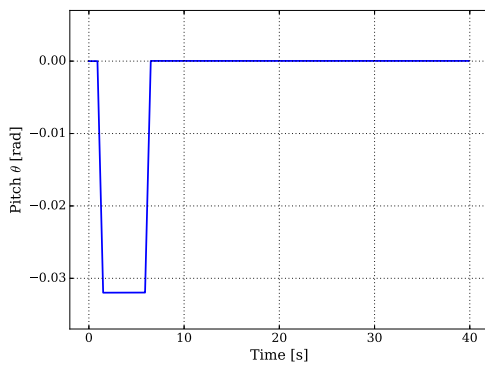
(b) Posición y .



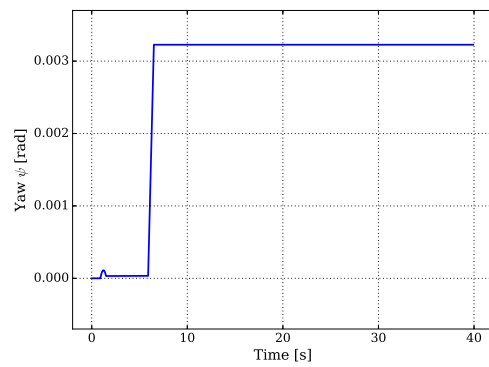
(c) Posición z .



(d) Ángulo de roll ϕ .



(e) Ángulo de pitch θ .



(f) Ángulo de yaw ψ .

Figura 2.10: Simulación en presencia de una rampa inclinada.

puede evidenciar que hay ocasiones en las que es beneficioso tener en cuenta las características del terreno y debido a esto, el modelo del robot debe tener en cuenta los 6 DOF a pesar de 3 de ellos estén impuestos por las características del terreno (altura, ángulo de *roll* y de *pitch*). Si el modelo del vehículo considera los 6 DOF, es posible incorporar las características del terreno y la orientación del vehículo en el diseño del controlador, lo que permite planificar las acciones de control de acuerdo a la superficie y la orientación del vehículo en vez de reaccionar a las mismas como si fuesen perturbaciones.

Capítulo 3

Navegación

3.1. Introducción

La integración de un sistema de navegación satelital (GNSS, Global Navigation Satellite System) con una unidad de medición inercial (IMU, Inertial Measurement Unit) es el estado del arte entre los sistemas de navegación [91, 127]. Los sensores inerciales han ganado terreno rápidamente en esta última década, principalmente debido a los significativos avances realizados en materia de microelectrónica [123]. El problema de la estimación de posición, velocidad y actitud de un cuerpo rígido en el espacio involucra ecuaciones de medición no lineales combinadas con la matriz de rotación, expresada a través de ángulos de Euler o cuaterniones, además de los modelos cinemáticos de la traslación y rotación del cuerpo rígido en el espacio. Tradicionalmente, los algoritmos utilizados para resolver el problema de la estimación de posición y actitud son el filtro de Kalman extendido (EKF, Extended Kalman Filter) [66, 76, 104], el Unscented Kalman Filter (UKF) [22, 100], o los más avanzados filtro de partículas (PF, Particle Filter) basados en la teoría Bayesiana de [15, 18].

Recientemente, la utilización de observadores no lineales se ha presentado como una alternativa interesante a los filtros de Kalman y los métodos estadísticos, sin embargo todavía existe poca literatura al respecto. Vanderveen et al. [127] utilizan un algoritmo de estimación basada en horizonte móvil (MHE, Moving Horizon Estimation) en tiempo real para estimar la actitud (u orientación) y los parámetros de calibración de los sensores de una nave espacial. En primer instancia, la actitud es estimada a partir de un magnetómetro y un giróscopo de tres ejes. Luego, utilizan un sensor estelar (star tracker) fusionado con los datos del magnetómetro y el giróscopo para obtener una estimación mejorada. Para lograr estimados en tiempo real, uti-

lizan un algoritmo de Gauss-Newton modificado para obtener la solución del problema de optimización; aprovechando la estructura de banda de la matriz resultante. En el trabajo de Poloni et al. [91] utilizan MHE para fusionar la información de un receptor GNSS con una IMU con el objetivo de estimar la posición, velocidad y actitud (u orientación) de un cuerpo en movimiento. En este trabajo, se utilizaron los datos obtenidos del vuelo de un avión de distintos sensores, algunos de bajo costo y otros de alta precisión. La combinación de un algoritmo de MHE junto con sensores GNSS e IMU de bajo costo logra mejor desempeño que el obtenido utilizando EKF con los mismos sensores. Asimismo, destacan que los resultados obtenidos mediante el uso de MHE y sensores de bajo costo es aproximado a los obtenidos utilizando EKF y sensores de alta precisión.

Tanto el EKF como MHE están basados en la solución de un problema de mínimos cuadrados. Mientras que EKF utiliza actualizaciones recursivas para obtener los estimados y la matriz de covarianza del error, MHE considera una ventana de horizonte finito y resuelve un problema de optimización sujeto a restricciones para encontrar los estimados óptimos (en el sentido de mínimos cuadrados). De esta manera, los límites físicos del sistema pueden ser modelados a través de cotas (o restricciones) en sus estados y parámetros. La omisión de esta información por parte del algoritmo de estimación puede degradar su performance [45]. Para sistemas lineales sin restricciones, este problema ha sido resuelto y varios métodos basados en distintas medidas estadísticas han sido desarrollados [21, 51]. Desafortunadamente, el filtro de Kalman no incorpora de forma explícita restricciones en los estimados (estados y parámetros) y debido a esto, se han desarrollado métodos *ad-hoc* para incorporar restricciones [116]. Los distintos métodos para incorporar restricciones al filtro de Kalman incluyen la reducción de modelo [117], proyección de los estimados [44], proyección de la ganancia [121], truncado de la función de densidad de probabilidad [118] y proyección del sistema [58]. Estos métodos conducen a soluciones subóptimas en el mejor de los casos y, bajo ciertas condiciones, pueden calcular estimados que no son realistas, sobre todo cuando las estadísticas de las variables desconocidas (estados iniciales, ruidos de medición y procesos, perturbaciones) se eligen de forma inadecuada. Por otro lado, la estimación de horizonte móvil (MHE) resuelve un problema de optimización para encontrar los estimados del sistema y de esta manera, provee un marco teórico para realizar estimación sujeta a restricciones.

Con cada muestra del sistema, MHE resuelve un problema de estimación de horizonte finito para determinar los estados y parámetros del sistema. Cuando una nueva medición se encuentra disponible, se descartan las mediciones viejas y el problema de estimación se vuelve a resolver para determinar los nuevos estimados (ver Figura 3.1). La información de las mediciones que

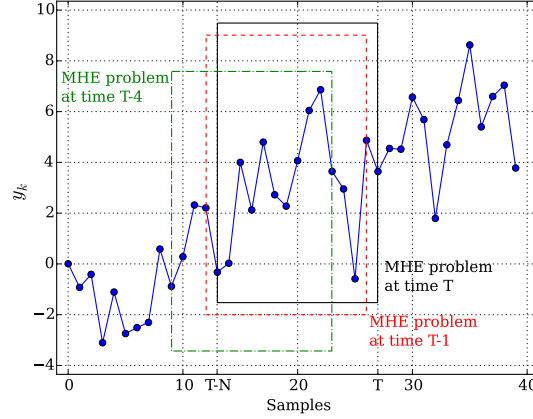


Figura 3.1: Esquema de actualización de MHE.

no son incluidas en la ventana de estimación es asimilada por la función objetivo a través de un término extra denominado *costo de arribo*. Este término caracteriza la distribución estadística de los estados al principio de la ventana de estimación dada la información proveniente de las mediciones. De esta manera, el costo de arribo permite que MHE aproxime a Mínimos Cuadrados y utilice un número finito de muestras. Una buena aproximación del costo de arribo permite reducir el tamaño de la ventana de estimación y por consiguiente, el del problema de optimización, mientras que mantiene buena performance y robustez. La forma más utilizada para aproximar el costo de arribo es utilizando un la norma-2 de los estados del principio de la ventana de estimación [19, 60, 93, 97]. Para sistemas lineales, Rao et al. [97] propusieron la actualización de los parámetros del costo de arribo (la matriz de peso y los estados iniciales) mediante la utilización de un filtro de Kalman. Dado que estos métodos utilizan distribuciones Gaussianas para modelar las densidad de probabilidad de los estados iniciales [96, 97], la aproximación del costo de arribo es inadecuada cuanto los estimados tienen restricciones y debido a esto, se obtienen malos estimados. Este problema surge debido a que las restricciones modifican la distribución de probabilidad de los ruidos y los estimados, forzando a cero la probabilidad de algunos valores y, en ocasiones, elimina la independencia entre los estimados y los ruidos [102].

Para hacer frente a estos problemas, Chu et al. [19] desarrollaron un método de actualización iterativo del costo de arribo que utiliza aproximaciones cuadráticas y la información de las restricciones activas e inactivas de la iteración previa. Estas ideas permiten construir aproximaciones cuadráticas en la proximidad de la solución óptima del costo de arribo exacto. El método

de actualización está basado en la hipótesis que el conjunto de restricciones activas no cambia después de un tiempo determinado. Esta hipótesis funciona cuando se eligen ventanas de estimación grandes. Sin embargo, si alguna restricción cambia su estado después de haber supuesto lo contrario, el estimador puede diverger. Además, este mecanismo pondera de más los datos pasados al retener la información de las restricciones, y de esta forma le resta importancia al efecto de la información disponible en los datos recientes, siendo incapaz de incorporarla en los nuevos estimados.

Finalmente, en un trabajo reciente Al-Matouq y Vincent [3] desarrollaron un algoritmo de MHE basado en múltiples ventanas de estimación. El algoritmo aprovecha las restricciones inactivas para reducir el tamaño del problema de optimización mientras que mantiene las propiedades de estabilidad y performance de mínimos cuadrados. Estas propiedades son preservadas mediante la aproximación del costo de arribo con un estimador sin restricciones que reformula la función objetivo en las regiones de restricciones inactiva muestra a muestra. Sin embargo, la utilización de múltiples ventanas incrementa la carga computacional, limitando la utilización de este método en procesos que no requieran períodos de muestreo cortos.

En este capítulo se propone utilizar estimación adaptativa en combinación con MHE para mitigar los efectos de las distribuciones de probabilidad no Gaussianas y las restricciones en los estimados. La matriz de peso del costo de arribo es actualizada utilizando algoritmos de estimación adaptativa en combinación con la solución de MHE en la muestra previa. Debemos destacar que esta forma de actualización del costo de arribo es consistente con las restricciones y cotas en los estados y los ruidos, mientras que incorpora un mecanismo de *feedback* entre las mediciones y los estimados, mejorando de esta manera el desempeño general del estimador. De esta forma, la matriz de covarianza es calculada a lazo cerrado en vez de a lazo abierto, tal como se calcula con las técnicas de estimación estándar. La eficiencia del enfoque propuesto será evaluada mediante simulación. Principalmente, se pretende mostrar que *i)* es importante tener en cuenta las cotas en los estimados cuando se actualiza el costo de arribo, *ii)* esta idea puede ser implementada utilizando técnicas de estimación adaptativa para aproximar el costo de arribo, *iii)* la estimación resultante de la combinación de MHE con el algoritmo propuesto es estable, y *iv)* el método de actualización propuesto permite reducir el tamaño de la ventana de estimación sin perder desempeño, lo que reduce el tamaño del problema de optimización. En las siguientes secciones se presentará el problema de MHE, para luego mostrar dos maneras diferentes de actualizar el costo de arribo. Se concluye presentando resultados en simulación.

3.2. Estimación basada en horizonte móvil lineal

Considere el sistema discreto, lineal e invariante en el tiempo

$$\begin{aligned} x_{k+1} &= Ax_k + Gw_k, \\ y_k &= Cx_k + v_k, \end{aligned} \quad (3.1)$$

donde $x_k \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ es el vector de estados, $w_k \in \mathcal{W} \subseteq \mathbb{R}^{n_w}$ es el vector de ruidos de proceso, $y_k \in \mathbb{R}^{n_y}$ es el vector de mediciones, $v_k \in \mathcal{V} \subseteq \mathbb{R}^{n_y}$ es el vector de ruidos de medición y $k \in \mathbb{I}$. Los ruidos \hat{w}_k y \hat{v}_k son perturbaciones estocásticas estacionarias, de media cero y momentos finitos. Los conjuntos de restricciones \mathcal{X} , \mathcal{W} y \mathcal{V} son considerados cerrados con $\mathbf{0} \in \mathcal{X}$, $\mathbf{0} \in \mathcal{W}$ y $\mathbf{0} \in \mathcal{V}$.

Mínimos cuadrados *batch* utiliza una secuencia de mediciones $\{y_j\}_{j=0}^k$ para estimar los estados \hat{x}_k que resuelven el siguiente problema de optimización

$$\begin{aligned} \min_{\hat{x}_{0|k}, \hat{\mathbf{w}}_k} \Phi_k &= \|\hat{x}_{0|k} - \bar{x}_0\|_{P_0^{-1}}^2 + \sum_{j=0}^k \|\hat{w}_{j|k}\|_{Q^{-1}}^2 + \|\hat{v}_{j|k}\|_{R^{-1}}^2 \\ \text{s.t.} \quad &\begin{cases} \hat{x}_{j+1|k} = A\hat{x}_{j|k} + G\hat{w}_{j|k}, \\ y_j = C\hat{x}_{j|k} + \hat{v}_{j|k}, \\ \hat{x}_{j|k} \in \mathcal{X}, \hat{w}_{j|k} \in \mathcal{W}, \hat{v}_{j|k} \in \mathcal{V}, \end{cases} \end{aligned} \quad (3.2)$$

donde $\|\cdot\|_K$ es la norma- K inducida, Q^{-1} y R^{-1} son matrices simétricas definidas positivas que penalizan los errores de los vectores estimados de ruido de proceso $\hat{w}_{j|k}$ y de medición $\hat{v}_{j|k}$ error. El par (\bar{x}_0, P_0^{-1}) resume la información previa a tiempo $k = 0$, donde \bar{x}_0 es el conocimiento actual del estado estimado inicial y P_0^{-1} es una matriz simétrica definida positiva. La solución al problema (3.2) calcula los estimados suavizados $\{\hat{x}_{j|k} : j = 0, 1, \dots, k-1\}$ y el estado filtrado $\hat{x}_{k|k}$.

Dado que mínimos cuadrados batch usa todas las mediciones disponibles, el tamaño del problema crece a medida que lleguen nuevas mediciones, convirtiéndolo en un problema computacionalmente intratable. MHE en cambio, considera una ventana fija de mediciones. A medida que una nueva medición llega, esta ventana se desliza un instante hacia adelante y el proceso se repite. Las mediciones pasadas son tenidas en cuenta mediante el término de penalidad Z_{k-N} denominado *costo de arribo* [97]. Entonces, el problema de MHE consiste en encontrar los estados, ruidos de proceso y ruidos de medición que

son solución del siguiente problema de optimización

$$\begin{aligned} \min_{\hat{x}_{k-N|k}, \hat{\mathbf{w}}_k} \Psi_k^N &= Z_{k-N}(\hat{x}_{k-N|k}) + \sum_{j=k-N}^k \|\hat{w}_{j|k}\|_{Q^{-1}}^2 + \|\hat{v}_{j|k}\|_{R^{-1}}^2 \\ \text{s.t.} \quad &\begin{cases} \hat{x}_{j+1|k} = A\hat{x}_{j|k} + G\hat{w}_{j|k}, \\ y_j = C\hat{x}_{j|k} + \hat{v}_{j|k}, \\ \hat{x}_{j|k} \in \mathcal{X}, \hat{w}_{j|k} \in \mathcal{W}, \hat{v}_{j|k} \in \mathcal{V}. \end{cases} \end{aligned} \quad (3.3)$$

La minimización de mínimos cuadrados batch (3.2) es equivalente a minimizar el problema de horizonte móvil (3.3) siempre que se cumpla la condición de que Z_{k-N} es el costo de arribo *exacto* Z_{k-N}^*

$$\begin{aligned} Z_{k-N}^* &= \min_{\hat{x}_{0|k-N}, \hat{\mathbf{w}}_{k-N}} \|\hat{x}_{0|k-N} - \bar{x}_0\|_{P_0^{-1}}^2 + \sum_{j=0}^{k-N} \|\hat{w}_{j|k-N}\|_{Q^{-1}}^2 + \|\hat{v}_{j|k-N}\|_{R^{-1}}^2 \\ \text{s.t.} \quad &\begin{cases} \hat{x}_{j+1|k-N} = A\hat{x}_{j|k-N} + G\hat{w}_{j|k-N}, \\ y_j = C\hat{x}_{j|k-N} + \hat{v}_{j|k-N}, \\ \hat{x}_{j|k-N} \in \mathcal{X}, \hat{w}_{j|k-N} \in \mathcal{W}, \hat{v}_{j|k-N} \in \mathcal{V}. \end{cases} \end{aligned} \quad (3.4)$$

El cálculo exacto del costo de arribo Z_{k-N}^* es complicado, ya que es una función cuadrática definida a trozos, donde la transición entre segmentos depende del estado de las restricciones (activas o inactivas) en el problema (3.2). Por ello, calcular Z_{k-N}^* requiere la solución de un problema de optimización cuadrático sujeto a restricciones (QP, del inglés Quadratic Program) cuyo tamaño crece con el horizonte $k - N$. En consecuencia, Z_{k-N}^* no puede ser calculado mediante la resolución del problema de optimización (3.4) y por ello, se debe utilizar una aproximación Z_{k-N} . Las formulaciones convencionales de MHE utilizan una ventana de estimación larga para reducir el efecto de los errores introducidos por una mala aproximación del costo de arribo [90]. En este caso, MHE usa toda la información disponible para el sistema, descartando la que no tiene efecto en la estimación dada la memoria finita del sistema. Si se quisiera achicar la ventana de estimación, es necesario tener mejores aproximaciones del costo de arribo de forma tal que incorpore la información que se descarta en el problema. Para sistemas lineales, generalmente se supone que la distribución de probabilidad es Gaussiana y la matriz de covarianza en el costo de arribo se actualiza utilizando un filtro de Kalman [85]. Es más, para sistemas sin restricciones que cumplen con la suposición Gaussiana, MHE equivale al filtro de Kalman. La matriz de penalidad P_{k-N} generalmente se calcula utilizando programación dinámica y de esta forma, se la puede interpretar como la covarianza del estado estimado a tiempo $k - N$ [97].

3.3. Actualización del costo de arribo

MHE se basa en representar el problema de mínimos cuadrados batch (3.2) en la forma (3.3), donde N se mantiene fijo. Entonces, para cada $k > N$, el costo de arribo Z_{k-N}^* es calculado utilizando la Ecuación (3.4) y luego se obtiene una nueva solución para el problema de MHE. Para evitar los inconvenientes generados por la dimensión del problema (3.4), éste puede ser reescrito de forma recursiva extendiendo la minimización parcial de (3.4) hacia la siguiente variable de decisión:

$$Z_{k-N}^* = \min_{\hat{\mathbf{w}}_k} Z_{k-N-1}^* + \|\hat{w}_{k-N-1|k}\|_{Q^{-1}}^2 + \|\hat{v}_{k-N|k}\|_{R^{-1}}^2$$

$$\text{s.t.} \begin{cases} \hat{x}_{k-N|k} = A\hat{x}_{k-N-1|k-1} + G\hat{w}_{k-N-1|k}, \\ y_{k-N} = C\hat{x}_{k-N|k} + \hat{v}_{k-N|k}, \\ \hat{x}_{k-N|k} \in \mathcal{X}, \hat{w}_{k-N-1|k} \in \mathcal{W}, \hat{v}_{k-N|k} \in \mathcal{V}. \end{cases} \quad (3.5)$$

Comenzando con

$$Z_0 = \|\hat{x}_0 - \bar{x}_0\|_{P_0^{-1}}^2, \quad (3.6)$$

dado que es muy difícil encontrar una expresión analítica exacta para calcular el costo de arribo de MHE con restricciones, se seguirá un enfoque como el de Rao et al. [97] y se utilizará una aproximación del costo de arribo basado en el costo de arribo del problema sin restricciones. De esta forma, el cálculo exacto del costo de arribo será aproximado por la aproximación cuadrática

$$\tilde{Z}_{k-N} = \|\hat{x}_{k-N} - \bar{x}_{k-N}\|_{P_{k-N}^{-1}}^2, \quad (3.7)$$

donde P_{k-N} y \bar{x}_{k-N} determinan completamente el costo de arribo aproximado. Por lo tanto, la actualización del costo de arribo aproximado Z_{k-N} puede formularse como una actualización de P_{k-N} y \bar{x}_{k-N} . Si se considera el problema de minimización parcial dado por la Ecuación (3.5) donde Z_{k-N-1} se obtiene mediante la Ecuación (3.7), resulta en un problema de estimación secuencial con restricciones.

Para problemas sin restricciones, la solución de la Ecuación (3.5) utilizando la Ecuación (3.7) conduce a un problema de estimación secuencial que puede ser resuelto utilizando un filtro de Kalman [85]. Con este enfoque, la información previa es transferida a la ventana de estimación actual utilizando el estado óptimo en la iteración anterior, $\hat{x}_{k-N-1|k-N-1}^*$ como estado inicial a tiempo k . Esta idea puede ser utilizada en problemas con restricciones siempre que se reemplace el filtro de Kalman por alguno de los métodos ad-hoc descritos en [116]. Sin embargo, las restricciones solo se aplican a los estados y no se tienen en cuenta para las matrices de covarianza P_{k-N} . El principal

argumento a favor de este tipo de actualización es que ninguna medición y_j es sobreponderada durante el proceso de estimación, lo que garantiza que no se utilice ninguna medición dos veces. Por otro lado, una de las desventajas de usar esta actualización es el hecho que exhibe un fenómeno periódico conocido como *ciclo de estimación*, asociado a confiar en estimados calculados a partir de pocos datos. El ciclo de estimación puede ser evitado utilizando una actualización con suavizado [122]. La actualización con suavizado incorpora más información en la actualización de \bar{x}_{k-N} . Otro problema de este tipo de actualizaciones son los errores de aproximación del costo de arribo cuando las restricciones cambian de estado entre estar activas o inactivas en el óptimo. Las restricciones modifican la distribución de probabilidad y los ruidos de manera que las fórmulas analíticas para la actualización no pueden utilizarse. En esta tesis, la actualización del costo de arribo aproximado (3.5) es formulada como una actualización de los términos P_{k-N} y \bar{x}_{k-N} de la Ecuación (3.7) y se realiza en dos pasos: *i*) actualizar \bar{x}_{k-N} y luego *ii*) actualizar P_{k-N} .

Cálculo de \bar{x}_{k-N} y P_{k-N}

Uno de los componentes de Z_{k-N} es el estado inicial \bar{x}_{k-N} . Dado que el problema de MHE calcula los estimados suavizados $\{\hat{x}_{j|k} : j = k-N, \dots, k-1\}$ y el estimado filtrado $\hat{x}_{k|k}$, podemos utilizar el estado estimado óptimo de la iteración anterior $\hat{x}_{k-N|k-1}^*$ como nuestro estado inicial,

$$\bar{x}_{k-N} = \hat{x}_{k-N|k-1}^*.$$

El uso del estado suavizado implica que se está haciendo una hipótesis similar a la de Chu et al. [19]: el estado de las restricciones (activas o inactivas) no cambia una vez que el estimado deja la ventana de estimación.

La matriz de penalidad P_{k-N} puede ser interpretada como la covarianza de x_{k-N} en el enfoque estocástico de MHE [97]. Existen soluciones analíticas para calcular P_{k-N} para problemas lineales sin restricciones. Para problemas lineales con restricciones (y para problemas no lineales) se han desarrollado soluciones que aproximan a P_{k-N} . Se basan en una actualización recursiva que emplea la información del sistema disponible a tiempo k (modelo del sistema, matriz de covarianza previa, restricciones activas, covarianza de los ruidos). Estas ecuaciones evolucionan a lazo abierto sin incorporar información de las mediciones o de los estados estimados.

En el ámbito del procesamiento de señales y estimación adaptativa, generalmente se utiliza una actualización recursiva basada en mediciones y/o los estados estimados [62]:

$$P_{k-N}^{-1} = \alpha_k P_{k-N-1}^{-1} + \beta_k \hat{x}_{k-N|k-1} \hat{x}_{k-N|k-1}^T \quad P_0 > 0, \quad (3.8)$$

donde $\alpha_k \in [0, 1)$ y $\beta_k \in (0, 2]$ son secuencias de peso variantes en el tiempo. La Ecuación (3.8) calcula de forma recursiva y en tiempo-real una estimación de la matriz de covarianza P_{k-N} al actualizar el estimado previo P_{k-N-1} con un promediado exponencial de $\hat{x}_{k-N|k-1}\hat{x}_{k-N|k-1}^T$. Este mecanismo de actualización puede ser interpretado como un filtro variante en el tiempo cuya entrada es $\hat{x}_{k-N|k-1}\hat{x}_{k-N|k-1}^T$ y la condición inicial P_0 .

Lema 3.1. Sean $\{A_i : i = 1, \dots, n\}$ matrices positivas definidas y sean $\{\gamma_i : i = 1, \dots, n\}$ números reales no negativos. Entonces $\sum_{i=1}^n \gamma_i A_i$ es positiva semidefinida.

Demostración. Sea $x \in \mathbb{R}^n$ distinto de cero, se puede ver que $x^T (\sum_{i=1}^n \gamma_i A_i) x = \sum_{i=1}^n \gamma_i (x^T A_i x) \geq 0$. Cada $\gamma_i \geq 0$ y cada $x^T A_i x > 0$. Esta suma resulta en una matriz positiva definida si alguno de los sumandos es positivo. \square

Lema 3.2. Si $P_0 > 0$ y $\alpha_0 > 0$, entonces la matriz P_k obtenida utilizando la ecuación (3.8) es positiva definida para todo $k \geq 0$.

Demostración. Dado que xx^T es simétrica positiva semidefinida, del Lema 3.1 se obtiene que P_k es definida positiva para todo $k \geq 0$. \square

Lema 3.3. Si $P > 0$, entonces existe una matriz invertible Z tal que $Z^T P Z = I$ y $Z Z^T = P^{-1}$.

Demostración. Ver Lema 7, página 261 de Johnson [54]. \square

Nótese que α_k y β_k tienen efectos contrarios. Cuando $\alpha_k \ll 1$, el algoritmo de adaptación tiende a descartar la información vieja resumida en P_{k-N-1} , estimando P_{k-N} con los datos recientes. Esto evita sobreponderar los datos viejos (reteniendo, por ejemplo, datos con restricciones activas) que pueden deteriorar el desempeño del estimador y logra mejorar la estimación de $\hat{x}_{k-N|k}$. Desde el punto de vista de la optimización, cuando el algoritmo descarta los datos que no contienen información útil, la matriz P_{k-N}^{-1} crece y la información inicial contenida en el costo de arribo tiende a ser más relevante. En cambio, cuando $\alpha_k \approx 1$, el algoritmo de adaptación tiende a retener la información resumida en P_{k-N-1} y descarta los datos actuales debido a que no incorporan información nueva. Esto permite incluir la información del pasado que es relevante al problema de estimación, mejorando así la estimación de $\hat{x}_{k-N|k}$ al incluir más información y olvidar solo los datos iniciales, lo que hace que P_{k-N}^{-1} decrezca. Cuando $\beta_k \gg 0$ el algoritmo de adaptación tiende a incrementar P_{k-N}^{-1} por la incorporación de información proveniente de los estimados $\hat{x}_{k-N|k-1}$. Esto permite incluir más información de los datos actuales en la matriz P_{k-N}^{-1} . Cuando $\beta_k \approx 0$ el algoritmo de adaptación descarta

la información disponible en $\hat{x}_{k-N|k-1}$ y solo aplica un mecanismo de olvido a P_{k-N}^{-1} , el cual depende del valor de α_k . En general, α_k y β_k permiten que el algoritmo de estimación descarte y/o incorpore información en la matriz P_{k-N}^{-1} de acuerdo a la información disponible en las mediciones $\{y_j\}_{j=k-N}^k$, los estados estimados $\hat{x}_{k-N|k}$ y el ruido de proceso $\{w_j\}_{j=k-N}^k$.

Si utilizamos el lema de inversión de matrices, la Ecuación (3.8) puede ser reescrita como

$$P_{k-N} = \frac{1}{\alpha_k} \left[P_{k-N-1} - \frac{P_{k-N-1} \hat{x}_{k-N|k-1} \hat{x}_{k-N|k-1}^T P_{k-N-1}}{\frac{\alpha_k}{\beta_k} + \hat{x}_{k-N|k-1}^T P_{k-N-1} \hat{x}_{k-N|k-1}} \right]. \quad (3.9)$$

Esta forma de calcular P_{k-N} incorpora un lazo de realimentación al problema de MHE, lo que permite el ajuste P_{k-N}^{-1} de acuerdo a la información disponible en las mediciones $\{y_j\}_{j=k-N}^k$ y debido a esto, se garantiza la convergencia de los estimados $\hat{x}_{k-N|k}$ y $\{w_j\}_{j=k-N}^k$. Debemos ser cuidadosos en la elección del mecanismo de actualización del costo de arriba, ya que malas aproximaciones pueden conducir a inestabilidad en el estimador [38, 95]. Por este motivo, en las secciones que siguen se estudiarán las condiciones que garantizan estabilidad y convergencia. Si la secuencia actual de ruido de proceso $\{w_j\}_{j=k-N}^k$ es grande, P_{k-N}^{-1} se incrementará, lo que hará que Z_{k-N} gane relevancia sobre los otros términos del problema de optimización. Eventualmente, $Z_{k-N} \rightarrow 0$ y mejorará la estimación en términos generales. En ese caso, P_{k-N}^{-1} decrecerá hasta que Z_{k-N} alcance su valor estacionario o la realimentación comience otra vez (see Fig 3.3b).

Se pueden obtener diferentes perfiles para el factor de olvido dependiendo de las características del problema y de los valores α_k y β_k [62]. Por ejemplo, si los estimados varían lentamente y las restricciones no cambian su estado una vez que salen de la ventana de estimación, la selección más apropiada es

$$0 < \alpha_k = \alpha < 1, \quad \beta_k = 1, \quad (3.10)$$

que conduce a un factor de olvido constante. En este caso, la Ecuación (3.9) estimará P_{k-N} utilizando los estimados disponibles de MHE, quitando relevancia a las observaciones pasadas a tasa constante. Cuando se utiliza un factor de olvido constante, la elección de α_k y β_k dados por la ecuación (3.10) conduce a una ganancia adaptativa decreciente, donde la información más vieja se olvida continuamente. Si el factor de olvido α no es elegido cuidadosamente, puede conducir a un crecimiento exponencial de la matriz P_{k-N} y el algoritmo de MHE se puede volver sumamente sensible a perturbaciones y/o errores numéricos [39].

Por este motivo, en muchos trabajos [8, 68, 89] se sugiere utilizar un factor de olvido variante en el tiempo, el cual permite que el algoritmo siga los

cambios lentos y también los repentinos que ocurran en el sistema, mientras que previene que la matriz P_{k-N} crezca y conduzca a errores numéricos. Esta elección es apropiada cuando los estimados son estacionarios y las restricciones pueden cambiar su estado una vez que salen de la ventana de estimación [62], ya que se debe olvidar los datos viejos con el fin de mantener la información relevante de las restricciones de los estados. A continuación se hará una reseña de algunos de los mecanismos existentes para actualizar los parámetros α_k y β_k . Por ejemplo si la evolución de α_k está dada por

$$\alpha_k = 1 - \frac{\hat{x}_{k-N|k-1}^T P_{k-N-1} \hat{x}_{k-N|k-1}}{1 + \hat{x}_{k-N|k-1}^T P_{k-N-1} \hat{x}_{k-N|k-1}}, \quad \beta_k = 1, \quad (3.11)$$

podemos ver que el factor de olvido de la ecuación (3.11) depende de los estimados, lo que conduce a que varíe en el tiempo. Si la norma de los estimados se vuelve nula, α_k vale automáticamente 1, lo que conduce al algoritmo de mínimos cuadrados recursivo (RLS, Recursive Least Squares). En los casos en que la secuencia de estados es tal que el término $\hat{x}_{k-N|k-1}^T P_{k-N-1} \hat{x}_{k-N|k-1}$ es significativo con respecto a 1, el factor de olvido toma un valor pequeño, lo que asegura buena capacidad de adaptación. El algoritmo proveerá una adaptación rápida y evitará el crecimiento exponencial de P_{k-N} , estabilizando así los estimados (controlando la cantidad de información empleada para calcular P_{k-N}) y el problema MHE.

Si los estimados son variantes en el tiempo y las restricciones pueden cambiar de estado luego de dejar la ventana de estimación, el factor de olvido debe garantizar una traza constante de la matriz de covarianza, que puede obtenerse mediante

$$\alpha_k = \frac{1}{\Xi} \text{trace} \left(P_{k-N-1} - \frac{P_{k-N-1} \hat{x}_{k-N|k-1} \hat{x}_{k-N|k-1}^T P_{k-N-1}}{\eta + \hat{x}_{k-N|k-1}^T P_{k-N-1} \hat{x}_{k-N|k-1}} \right), \quad \beta_k = \eta^{-1} \alpha_k, \quad (3.12)$$

donde $\Xi > 0$ es la traza deseada de la matriz de covarianza y $\eta > 0$ es una constante elegida de forma adecuada para cada problema.

Fortescue et al. [39] propusieron un algoritmo de olvido variable que trata de mantener la cantidad de información constante y muestran que una elección razonable de la cantidad de información puede prevenir que la matriz de peso crezca demasiado y a la vez, se mantenga la capacidad de adaptación. Posteriormente, Osorio Cordero y Mayne [89] modificaron el algoritmo para asegurar convergencia global. Las modificaciones realizadas garantiza que la que trace P_k (y por lo tanto, $\|P_k\|$) es siempre menor que una constante c . El algoritmo calcula el estimado mediante RLS si $\alpha_k = 1$, y mediante mínimos cuadrados ponderado en el caso que $\alpha_k \in (0, 1)$. La longitud de la memoria

N_k está relacionada con el factor de olvido α_k mediante $\alpha_k = 1 - 1/N_k$; factores de olvido de 0, 0.99 y 1 corresponden a longitudes de memoria de 1, 100 y ∞ , respectivamente. Las demás constantes requeridas por el algoritmo pueden ser elegidas de la siguiente manera: σ se elige de manera que σ/σ_w sea grande, donde σ_w denota la varianza del ruido de proceso; $P_0 = \delta I$, donde δ es grande (10^6 , por ejemplo), $c > \delta$ es grande. El factor de olvido variable y la matriz de información pueden ser calculados mediante

$$\begin{aligned}
 \epsilon_k &= y_{k-N} - \hat{y}_{k-N} \\
 N_k &= [1 + \hat{x}_{k-N|k-1}^T P_{k-N-1} \hat{x}_{k-N|k-1}] \frac{\sigma}{\|\epsilon_{k-N}\|_2^2} \\
 \alpha_k &= 1 - 1/N_k \\
 W_k &= [I - \frac{P_{k-N-1} \hat{x}_{k-N|k-1} \hat{x}_{k-N|k-1}^T}{1 + \hat{x}_{k-N|k-1}^T P_{k-N-1} \hat{x}_{k-N|k-1}}] P_{k-N-1} \\
 P_{k-N} &= \frac{1}{\alpha_k} W_k, \text{ si } 1/\alpha_k \text{ trace}(W_k) \leq c \\
 P_{k-N} &= W_k, \text{ en otro caso.}
 \end{aligned} \tag{3.13}$$

Observación 3.1. *El algoritmo de estimación de la matriz de covarianza puede expresarse como un filtro de Kalman con el estimado suavizado $\hat{x}_{k-N|k-1}$ como estado estimado, \hat{v}_k como el residuo de estimación y P_{k-N} como la matriz de covarianza [62]. En el caso de estimación sin restricciones, el algoritmo de estimación de la matriz de covarianza equivale a una actualización mediante filtrado [85]. La relación entre RLS y el filtro de Kalman es conocida [51]. En el caso del filtro de Kalman, la formula de actualización de la matriz de covarianza es un caso especial de la ecuación (3.8) bajo suposiciones específicas y una elección particular de α_k y β_k . Para una descripción detallada de estas equivalencias, el lector puede ver la Sección 3.2.4 de Landau et al. [62]*

Resumen de MHE con actualización adaptativa del costo de arribo

MHE resuelve el problema (3.3), obteniendo una solución $\{\hat{x}_{j|k}\}_{j=k-N}^k$ que aproxima la solución óptima del problema (3.2). Para $k \leq N$, se resuelve el problema (3.2) de mínimos cuadrados batch. Luego, cuando $k > N$, se resuelve el problema MHE (3.4) con el costo de arribo aproximado

$$\tilde{Z}_{k-N} = \left\| \hat{x}_{k-N|k} - \hat{x}_{k-N|k-1}^* \right\|_{P_{k-N}^{-1}}^2, \tag{3.14}$$

cuya solución puede ser obtenida utilizando cualquier *solver* QP. La matriz P_{k-N} y el estimado del estado inicial \bar{x}_{k-N} son actualizados y la ventana avanza una muestra, actualizando así el vector de mediciones $\{y_j\}_{j=k-N}^k$. El algoritmo [1](#) resume el estimador de horizonte móvil propuesto. El método adaptativo utilizado para actualizar la matriz de covarianza define el comportamiento del algoritmo. Si se desea utilizar un enfoque de traza constante (MHE_{AD-CT}), se utiliza la Ecuación [\(3.12\)](#). En cambio, si se desea utilizar un factor de olvido variable (MHE_{AD-VF}), se utiliza la Ecuación [\(3.11\)](#).

Algoritmo 1: Algoritmo MHE_{AD}

$\bar{x}_0, P_0 > 0, Q, R > 0$ and $N \geq 1$

Inicialización:

$\bar{x}_{k-N} \leftarrow \bar{x}_0$

$P_{k-N} \leftarrow P_0$

for muestras $k = 0, 1, 2, \dots$ **do**

Obtener medición y_k

if $k < N$ **then**

Preparar el vector de datos:

$Y \leftarrow [y_0, \dots, y_k]^T$

Resolver Φ_k con \bar{x}_0, P_0 y Y

else

Prepare data vector:

$Y \leftarrow [y_{k-N}, \dots, y_k]^T$

Resolver Ψ_k^N con \bar{x}_{k-N}, P_{k-N} y Y

$\bar{x}_{k-N} \leftarrow \hat{x}_{k-N+1|k}$

Actualizar P_{k-N} usando la Ec. [\(3.12\)](#) o [\(3.13\)](#)

end if

Obtener los estimados actuales $\hat{x}_{j|k}$ para $j = k - N, \dots, k$

end for

3.4. Análisis de estabilidad

La estabilidad de un estimador implica que el residuo converge a cero para el sistema nominal

$$\begin{aligned} x_{k+1} &= Ax_k, \\ y_k &= Cx_k, \end{aligned} \tag{3.15}$$

sin ruido de proceso o de medición. Sin embargo, para el caso de estimación con restricciones, una mala elección de las restricciones puede evitar la

convergencia al estado real del sistema. Rao et al. [97] agregaron el requerimiento de que la evolución del sistema respete las restricciones, las cuales solo necesitan satisfacer la siguiente suposición para demostrar estabilidad.

Suposición 3.1. *Suponga que el sistema (3.15) con condición inicial x_0 genera los datos. Se supondrá que existe $x_{0|\infty}$, $\{w_{k|\infty}\}_{k=0}^{\infty}$ y $\sigma > 0$ tal que*

$$\sum_{k=0}^{\infty} \|v_{k|\infty}\|_{R^{-1}} + \|w_{k|\infty}\|_{Q^{-1}} + \|x_{0|\infty} - \hat{x}_0\|_{P_0^{-1}} \leq \sigma \|x_0 - \hat{x}_0\|^2$$

y $x_{k|\infty} \in \mathcal{X}$, $w_{k|\infty} \in \mathcal{W}$, $v_{k|\infty} \in \mathcal{V}$.

La Suposición 3.1 asegura la existencia de una trayectoria de estados y perturbaciones factible que genera un costo acotado. A continuación se estudiará la estabilidad de mínimos cuadrados batch (FIE, Full Information Estimator).

Observación 3.2. *Suponga que las matrices Q , R y P_0 son positivas definidas, (C, A) es observable, y se cumple la Suposición 3.1. Entonces el FIE con restricciones es un observador asintóticamente estable para el sistema (3.15).*

Demostración. Ver Muske et al. [85]. □

Observación 3.3. *La matriz P_k satisface $\|P_k\| \leq c, \forall k$. El hecho que P_k permanezca acotada, garantiza que el problema de optimización de MHE con actualización adaptativa del costo de arriba es convexo y dada la Suposición 3.1, existe una solución.*

Demostración. Ver Apéndice A.1. □

Observación 3.4. *La matriz P_k tiene solución de estado estacionario, es decir*

$$\lim_{k \rightarrow \infty} P_k = P_{\infty}$$

Demostración. Ver Apéndice A.2. □

Teorema 3.1. *$V_k = \tilde{x}_k^T P_k^{-1} \tilde{x}_k$ es una función de Lyapunov para MHE con actualización adaptativa del costo de arriba, donde $\tilde{x}_k = x_k - \hat{x}_k$ y la matriz P_k^{-1} están dados por la ecuación (3.13), es decir, $P_k^{-1} = \alpha_k(P_{k-1}^{-1} + \hat{x}_{k-N|k-1} \hat{x}_{k-N|k-1}^T)$.*

Demostración. Ver apéndice A.3. □

3.4.1. Estabilidad acotada

De la Observación [3.3](#) sabemos que $\|P_k\|$ es no creciente, por lo tanto $\|P_k^{-1}\|$ es no decreciente. Por la convergencia de P_k a P_∞ , tenemos que $\|P_k^{-1}\| \leq \|P_{k+1}^{-1}\| \leq \|P_\infty^{-1}\|$. Por lo tanto, el costo de arribo y cada uno de los costos de cada etapa satisfacen las siguientes cotas

$$\lambda_{\min}(P_{k-N}^{-1})\|\hat{x}_{k-N|k} - \bar{x}_{k-N}\| \leq \|\hat{x}_{k-N|k} - \bar{x}_{k-N}\|_{P_{k-N}^{-1}} \leq \lambda_{\max}(P_\infty^{-1})\|\hat{x}_{k-N|k} - \bar{x}_{k-N}\|$$

$$\lambda_{\min}(Q^{-1})\|\hat{w}\| \leq \|\hat{w}\|_{Q^{-1}} \leq \lambda_{\max}(Q^{-1})\|\hat{w}\|$$

$$\lambda_{\min}(R^{-1})\|\hat{v}\| \leq \|\hat{v}\|_{R^{-1}} \leq \lambda_{\max}(R^{-1})\|\hat{v}\|.$$

Bajo ruidos acotados, el estimador MHE con actualización adaptativa del costo de arribo es Robust Global Asymptotically Stable (RGAS). Es más, si $\lim_{k \rightarrow \infty} w_k = 0$ y $\lim_{k \rightarrow \infty} v_k = 0$, entonces $\lim_{k \rightarrow \infty} \|\hat{x}_k - \bar{x}_k\| = 0$.

Demostración. Ver Teorema 7 de [\[83\]](#). □

3.5. Resultados

Para las simulaciones se utilizará el modelo lineal discreto de Rao et al. [\[98\]](#)

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 0.99 & 0.2 \\ -0.1 & 0.3 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_k, \\ y_k &= \begin{bmatrix} 1 & -3 \end{bmatrix} x_k + v_k, \end{aligned} \tag{3.16}$$

donde $\{v_k\}$ es una secuencia de variables aleatorias independientes, con distribución normal, media cero y covarianza σ_v , y $w_k = |z_k|$ donde $\{z_k\}$ es una secuencia de variables aleatorias independientes, con distribución normal, media cero y covarianza σ_z . Para todos los ejemplos, el estado inicial \bar{x}_0 es una variable aleatoria con distribución normal, media cero y covarianza $\sigma_x = 0.5$, además, supondremos que $x_0 = [0, 0]^T$ y $\bar{x}_0 = [0.5, -0.5]$. El problema de estimación con restricciones es formulado con $Q = 1$, $R = 0.01$, $P_0 = 0.5$ y agregamos la desigualdad $w_k \geq 0$ para incluir la información disponible acerca de dicha secuencia aleatoria.

En primer lugar, los algoritmos propuestos (MHE_{AD-VF} y MHE_{AD-CT}) se comparan con FIE y con un algoritmo para estimación lineal (MHE_{KF}) disponible en las librerías *Nonlinear Model Predictive Control Tools for CasADi* (*mpc-tools-casadi*) [\[101\]](#). Este algoritmo propaga el costo de arribo y la estimación del estado inicial utilizando un filtro de Kalman. Las simulaciones

	$x^{(1)}$			
	FIE	MHE _{AD-VF}	MHE _{AD-CT}	MHE _{KF}
$N = 3$	13.22	20.44	27.37	103.42
$N = 6$	13.22	17.77	16.25	50.68
$N = 10$	13.22	15.28	14.51	26.54

Tabla 3.1: Comparación del efecto del tamaño del horizonte de estimación en la sumatoria del error cuadrático medio del estado $x^{(1)}$ para el sistema (3.16).

evalúan el efecto del tamaño del horizonte de estimación en la performance de los estimadores. Se utilizaron tres tamaños de horizontes ($N = 3$, $N = 6$ y $N = 10$) para cada uno de los algoritmos MHE con $\sigma_v = 0.1$ y $\sigma_w = 1.0$. Luego se calculó la suma del error cuadrático de estimación

$$\xi^{(i)} = \sum_{j=0}^k (x_j^{(i)} - \hat{x}_j^{(i)})^2,$$

donde el superíndice (i) denota la componente i -ésima de x . En esta prueba se realizaron 100 pruebas, donde se calculó el promedio de la suma del error cuadrático de estimación. Los resultados obtenidos se pueden ver en las Tablas 3.1 y 3.2. Se puede observar que los errores obtenidos por MHE_{AD} son siempre menores que los obtenidos por MHE_{KF} y a la vez, más cercanos a FIE para cada uno de los tamaños de horizonte considerados. A medida que el tamaño del horizonte es mayor, el costo de arribo tiene un rol menos relevante en la función de costo del problema de optimización y es por eso que se puede apreciar que el costo de todos los estimadores tienden a converger al mismo valor. Una buena aproximación del costo de arribo nos permite reducir el tamaño del horizonte. Teniendo esto en cuenta, podemos ver que los errores de MHE_{AD} no cambian de manera significativa con los distintos horizontes, mostrando así que el algoritmo propuesto parece ser un buen método para aproximar el costo de arribo.

Ejemplo 1: $\sigma_w = 1.0$, $\sigma_v = 0.1$, $N = 5$

En este ejemplo, la covarianza del ruido de proceso es $\sigma_w = 1.0$ y la covarianza del ruido de medición $\sigma_v = 0.1$. La Figura 3.2 muestra los estados $x^{(1)}$, $x^{(2)}$ y sus estimados (ver Fig. 3.2a y 3.2b) con un horizonte de $N = 5$. Se puede observar que los estimados de MHE_{AD} están más cerca de FIE a

	$x^{(2)}$			
	FIE	MHE _{AD-VF}	MHE _{AD-CT}	MHE _{KF}
$N = 3$	1.55	2.20	2.84	11.55
$N = 6$	1.55	1.96	1.76	5.71
$N = 10$	1.55	1.71	1.61	3.02

Tabla 3.2: Comparación del efecto del tamaño del horizonte de estimación en la sumatoria del error cuadrático medio del estado $x^{(2)}$ para el sistema (3.16).

pesar de la reducida longitud del horizonte. A medida que el tiempo pasa, los estimados de MHE_{AD} convergen a los estimados de FIE. Las Figuras 3.2c y 3.2d muestran el error cuadrático $\xi^{(i)}$ a cada tiempo de muestreo. En estas figuras se puede observar que el error de los algoritmos propuestos es similar al obtenido utilizando FIE. Por otro lado, en estas figuras también se puede observar que el desempeño de MHE_{KF} es peor, principalmente debido a que el tamaño del horizonte fue configurado en $N = 5$. Esta diferencia puede ser explicada por el hecho que la actualización del filtro de Kalman no aproxima adecuadamente el costo de arriba debido al efecto de las restricciones [102]. Esta situación empeora cuando el tamaño del horizonte es más chico, como se ha mencionado en el ejemplo anterior.

Ejemplo 2: $\sigma_w = 0.4$, $\sigma_v = 0.1$, $N = 5$

En este ejemplo, a pesar que la covarianza del ruido de proceso es $\sigma_w = 0.4$, varía con el tiempo y los distintos algoritmos MHE no tienen conocimiento de que esto sucede. La covarianza σ_w tendrá valores de 0.1 cuando $0 \leq k < 20$, 0.25 cuando $20 \leq k < 40$, 1.5 cuando $40 \leq k < 100$, 1.0 cuando $100 \leq k < 150$ y 0.4 cuando $150 \leq k < 200$. Podemos ver en la Fig. 3.3a que los algoritmos propuestos en este trabajo tienen mejor desempeño que MHE_{KF}. La Fig. 3.3b muestra la traza de la matriz P_{k-N}^{-1} utilizada por cada uno de los algoritmos. Podemos ver que la matriz de MHE_{KF} converge a un valor de estado estacionario y luego no cambia, mientras que los algoritmos propuestos adaptan el peso de esta matriz acorde a las condiciones cambiantes del ruido de proceso. En estos casos, la traza de P_{k-N}^{-1} crece cuando el error es pequeño, mostrando que tiene mas confianza en los estimados. Por otro lado, cuando la diferencia entre el estado real y el estimado crece, la traza de la matriz decrece; logrando así mejor adaptación, aún bajo la influencia de perturbaciones desconocidas y cambiantes. En las figuras 3.4a y

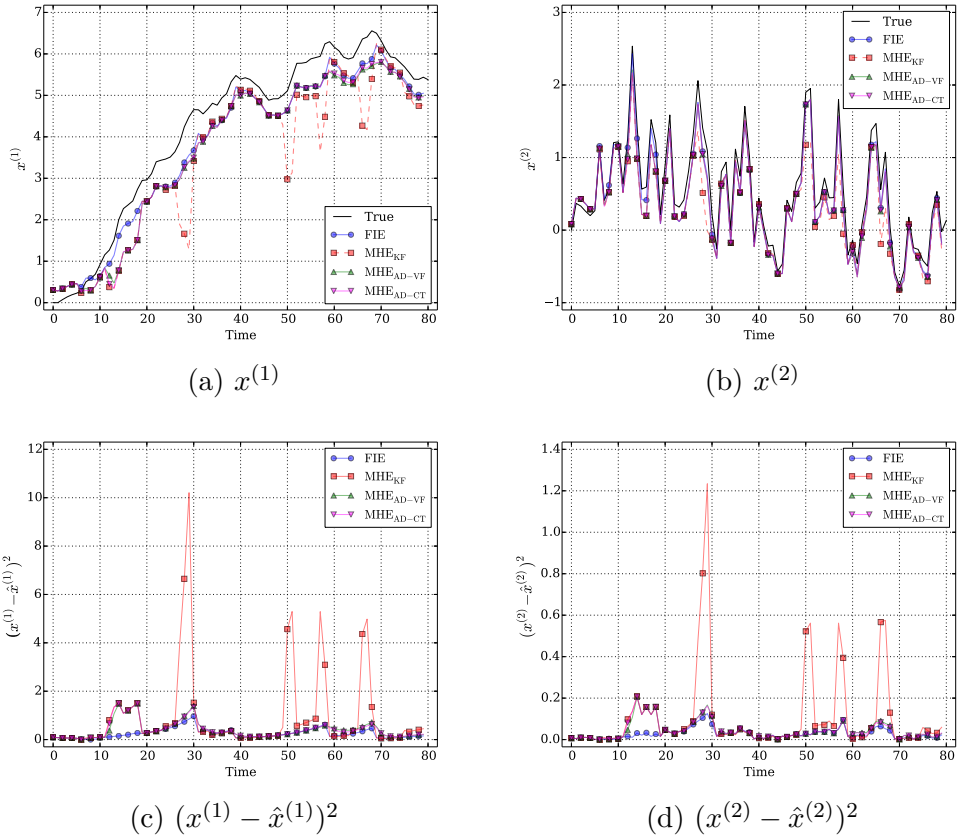


Figura 3.2: Comparación de los estimadores para el modelo (3.16) con horizonte de $N = 5$.

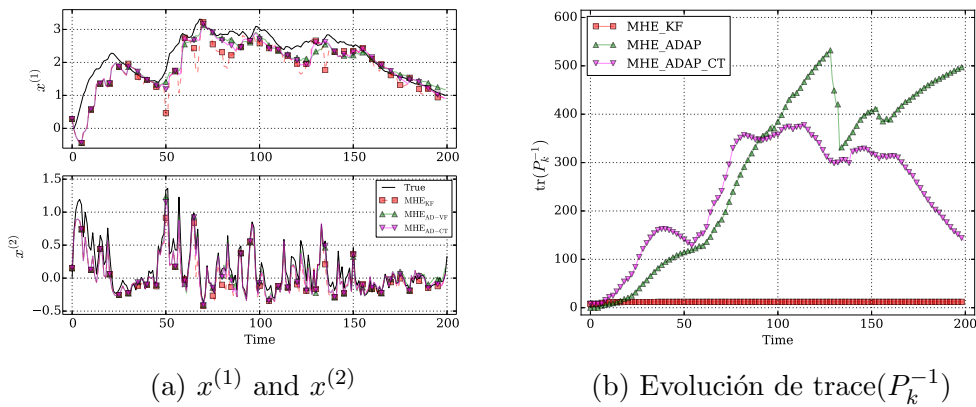


Figura 3.3: Comparación de los estimadores para el modelo (3.16) con horizonte de $N = 5$ y σ_w variable.

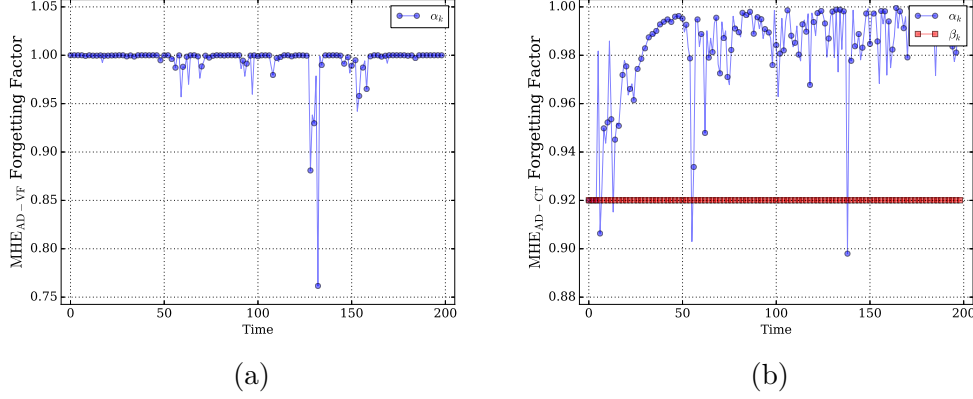


Figura 3.4: Evolución de las secuencias de factores de olvido para MHE_{VF} y MHE_{CT}.

3.4b se puede ver la evolución de los factores de olvido variable de los algoritmos MHE_{VF} y MHE_{CT}, respectivamente. En ambos casos se puede ver que α_k toma un valor muy cercano a 1 y el algoritmo se comporta como un estimador de mínimos cuadrados ponderado con una gran memoria. En los casos en los que el ruido crece y el estimador es incapaz de seguir al sistema, el factor de olvido decrece con el fin de permitir una adaptación rápida de la matriz de peso. Esta es la razón principal por la que el algoritmo propuesto logra una mejor aproximación del costo de arribo.

3.6. Estimación basada en horizonte móvil no lineal

Considere ahora el sistema discreto, no lineal

$$\begin{aligned} x_{k+1} &= f(x_k, w_k), \\ y_k &= h(x_k) + v_k \end{aligned} \quad (3.17)$$

donde $x_k = x(k) \in \mathcal{X} \subseteq \mathfrak{R}^{n_x}$ es el vector de estados, $w_k = w(k) \in \mathcal{W} \subseteq \mathfrak{R}^{n_w}$ y $v_k = v(k) \in \mathcal{V} \subseteq \mathfrak{R}^{n_v}$ son los ruidos de proceso y de medición, respectivamente y se supone que son secuencias de variables aleatorias independientes con funciones de densidad de probabilidad invariantes en el tiempo, y $y_k = y(k) \in \mathfrak{R}^{n_y}$ es el vector de medición.

Dado un horizonte de N mediciones $\{y_{k-N}, \dots, y_k\}$ hasta tiempo k , el problema MHE no lineal consiste en encontrar los estados, ruidos de proceso

y de medición que resuelven el siguiente problema de optimización

$$\begin{aligned} \min_{\hat{x}_{k-N|k}, \hat{\mathbf{w}}_k} \mathcal{J}_k(\hat{x}_{k-N|k}, \hat{\mathbf{w}}_k) &= Z_{k-N}(\hat{x}_{k-N|k}) + \sum_{j=k-N}^k \|\hat{w}_{j|k}\|_{Q^{-1}}^2 + \|\hat{v}_{j|k}\|_{R^{-1}}^2 \\ \text{st.} \quad &\begin{cases} \hat{x}_{j+1|k} = f(\hat{x}_{j|k}, \hat{w}_{j|k}), \\ y_j = h(\hat{x}_{j|k}) + \hat{v}_{j|k}, \\ \hat{x}_{j|k} \in \mathcal{X}, \hat{w}_{j|k} \in \mathcal{W}, \hat{v}_{j|k} \in \mathcal{V}. \end{cases} \end{aligned} \quad (3.18)$$

El problema de optimización no lineal puede ser resuelto utilizando estrategias de *single shooting*, *multiple shooting*, *collocation* [29] o linealizaciones sucesivas [84]. En esta sección el enfoque trata en la aproximación del costo de arribo y el problema de optimización será resuelto utilizando una estrategia de *multiple shooting* sobre el problema no lineal.

Actualización del costo de arribo

Como se ha mencionado en la sección anterior, MHE reduce el cálculo necesario para FIE al resolver el problema de optimización sobre un horizonte finito de tamaño N . Para poder obtener buenos estimados con horizontes cortos, el algoritmo MHE debe ser capaz de obtener buenas aproximaciones del costo de arribo. De lo contrario, dependiendo del intervalo de muestreo y la respuesta dinámica del sistema, el tamaño del horizonte puede resultar sumamente largo.

Ya sea que los sistemas sean lineales o no, el método más utilizado para aproximar el costo de arribo es la norma 2 ponderada. Para el caso particular de sistemas no lineales, Rao y Rawlings [96] utilizaron una aproximación del costo de arribo basada en el EKF y propagaban el estado inicial una muestra hacia adelante utilizando la matriz de ganancia de dicho filtro. Tenny y Rawlings [122] compararon la estrategia basada en EKF utilizando filtrado y suavizado de los estimados, donde observaron que la estrategia de filtrado introducía oscilaciones no deseadas (*ciclo de estimación*). A pesar de su popularidad, una de las principales desventajas de los métodos de aproximación del costo de arribo basado en EKF es la suposición de que la distribución de probabilidad de los estados está bien representada mediante una Gaussiana. Si el sistema en consideración es no lineal, a pesar de que los ruidos de procesos y medición se supongan Gaussianos, las restricciones en el sistema o la propagación de las variables aleatorias por las ecuaciones del modelo hará que dejen de serlo. Por este motivo, muchos trabajos se han enfocado en aproximar el costo de arribo utilizando técnicas de filtrado que reconstruyen la distribución de probabilidad basándose en muestras de los estados.

Qu y Hahn [93] propusieron utilizar el UKF para actualizar sólo la matriz de covarianza del costo de arribo. Ungarala [124] compara el desempeño de varios filtros de este tipo para aproximar el costo de arribo. En su trabajo, el EKF es comparado con el UKF, el filtro de partículas y filtros basados en cadenas de Markov. Aquí se muestran ejemplos de simulación donde se observan los beneficios de este tipo de aproximaciones sobre el enfoque tradicional del EKF. López-Negrete, Patwardhanb, y Biegler [70] utilizan un filtro de partículas *con restricciones*. En este trabajo se muestra que la incorporación de las restricciones del sistema en el proceso de aproximación del costo de arribo conduce a mejoras en la estimación.

De esta forma, teniendo en cuenta la aproximación en norma 2, podemos decir que los filtros basados en muestreo se utilizan para generar los primeros dos momentos de la distribución del costo de arribo. Por ese motivo, en esta sección se utilizarán los métodos de aproximación adaptativa del costo de arribo en sistemas no lineales. El desempeño del método de olvido variable dado por la ecuación 3.13 se comparará con la aproximación del costo de arribo utilizando EKF. El motivo de esta elección se basa en el trasfondo práctico de este trabajo. Los filtros basados en muestreo necesitan una masa de datos mayor para poder obtener sus estimados, lo que conduce a una mayor carga computacional y debido a ello, no son muy utilizados en sistemas que requieran operación en tiempo real.

3.7. Resultados

Para las simulaciones se utilizará el modelo no lineal discreto de Rao et al. [98]

$$\begin{aligned} x_{k+1}^{(1)} &= 0.99 x_k^{(1)} + 0.2 x_k^{(2)} \\ x_{k+1}^{(2)} &= -0.1 x_k^{(1)} + \frac{0.5 x_k^{(2)}}{1 + (x_k^{(2)})^2} + w_k \\ y_k &= x_k^{(1)} - 3 x_k^{(2)} + v_k \end{aligned} \quad (3.19)$$

donde $\{v_k\}$ es una secuencia de variables aleatorias independientes, con distribución normal, media cero y covarianza σ_v , y $w_k = |z_k|$ donde $\{z_k\}$ es una secuencia de variables aleatorias independientes, con distribución normal, media cero y covarianza σ_w . Para todos los ejemplos, el estado inicial \bar{x}_0 es una variable aleatoria con distribución normal, media cero y covarianza $\sigma_x = 0.5$, además, supondremos que $x_0 = [1, -1]^T$ y $\bar{x}_0 = [0.5, -0.5]^T$. El problema de estimación con restricciones es formulado con $Q = 1$, $R = 0.01$. Se agrega la desigualdad $w_k \geq 0$ al problema de optimización para incluir la

	$x^{(1)}$		
	FIE	MHE _{AD-VF}	MHE _{EKF}
$N = 3$	37.09	175.59	253.17
$N = 6$	37.09	110.40	129.33
$N = 10$	37.09	66.58	72.84

Tabla 3.3: Comparación del efecto del tamaño del horizonte de estimación en la sumatoria del error cuadrático medio del estado $x^{(1)}$ para el sistema (3.19).

información disponible acerca de dicha secuencia aleatoria. Para el caso de la actualización con el filtro de Kalman Extendido, $P_0 = 4 I$ y para el caso de actualización adaptativa con olvido variable, $P_0 = 1000 I$.

Se procederá de la misma manera que para el caso lineal. Se comparará el algoritmo MHE_{AD-VF} con FIE y con MHE_{KF}, el cual utiliza el filtro de Kalman extendido para propagar el costo de arribo y el estado inicial. Las simulaciones evalúan el efecto del tamaño del horizonte de estimación en la performance de los estimadores. Se utilizaron tres tamaños de horizontes ($N = 3$, $N = 6$ y $N = 10$) para cada uno de los algoritmos MHE con $\sigma_v = 0.1$ y $\sigma_w = 1.0$. Luego se calculó la suma del error cuadrático de estimación utilizando la ecuación (3.17). Los resultados obtenidos se pueden ver en las Tablas 3.3 y 3.4, donde podemos observar que los errores obtenidos por MHE_{AD} son menores que los obtenidos por MHE_{KF} y a la vez, más cercanos a FIE para cada uno de los tamaños de horizonte considerados. A medida que el tamaño del horizonte es mayor, el costo de arribo tiene un rol menos relevante en la función de costo del problema de optimización y es por eso que se puede apreciar que el costo de todos los estimadores tienden a converger al mismo valor y podemos ver que los errores de MHE_{AD} no cambian de manera significativa con los distintos horizontes, mostrando así que el algoritmo propuesto parece ser un buen método para aproximar el costo de arribo. En el caso no lineal también queda claro que una buena aproximación del costo de arribo nos permite reducir el tamaño del horizonte. Y es en este caso donde esta propiedad es aún más relevante, dado que los problemas de optimización no lineales no sólo son más costosos computacionalmente que su parte lineal, sino que es más complicado garantizar su convexidad.

	$x^{(2)}$		
	FIE	MHE _{AD-VF}	MHE _{KF}
$N = 3$	4.29	16.54	28.18
$N = 6$	4.29	10.96	14.48
$N = 10$	4.29	7.19	8.24

Tabla 3.4: Comparación del efecto del tamaño del horizonte de estimación en la sumatoria del error cuadrático medio del estado $x^{(2)}$ para el sistema (3.19).

Ejemplo 3: $\sigma_w = 1.0$, $\sigma_v = 0.1$, $N = 5$

En este ejemplo, la covarianza del ruido de proceso es $\sigma_w = 1.0$ y la covarianza del ruido de medición $\sigma_v = 0.1$. La Figura 3.5 muestra los estados $x^{(1)}$, $x^{(2)}$ y sus estimados (ver Fig. 3.5a y 3.5b) con un horizonte de $N = 5$. Se puede observar que los estimados de MHE_{AD} están más cerca de FIE a pesar de la reducida longitud del horizonte. A medida que el tiempo pasa, los estimados de MHE_{AD} convergen a los estimados de FIE. Las Figuras 3.2c y 3.2d muestran el error cuadrático $\xi^{(i)}$ a cada tiempo de muestreo. En estas figuras se puede observar que el error de los algoritmos propuestos es similar al obtenido utilizando FIE. Por otro lado, en estas figuras también se puede observar que el desempeño de MHE_{KF} es peor. Esta diferencia puede ser explicada por el hecho que el filtro de Kalman actualiza el costo de arribo utilizando solamente información del modelo del sistema y no los datos. Si se utiliza una matriz de pesos Q o R incorrecta, la propagación de la matriz de covarianza cambia de forma significativa. Además, una vez que la matriz de peso P_k obtenida utilizando el filtro de Kalman extendido alcanza un estado estacionario, es incapaz de adaptarse a cambios del sistema.

Ejemplo 4: $\sigma_w = 0.4$, $\sigma_v = 0.1$, $N = 5$

En este ejemplo, a pesar que la covarianza del ruido de proceso es $\sigma_w = 0.4$, varía con el tiempo y los distintos algoritmos MHE no tienen conocimiento de que esto sucede. La covarianza σ_w tendrá valores de 0.1 cuando $0 \leq k < 20$, 0.25 cuando $20 \leq k < 40$, 1.5 cuando $40 \leq k < 100$, 1.0 cuando $100 \leq k < 150$ y 0.4 cuando $150 \leq k < 200$. En las figuras 3.6a y 3.6b podemos ver la evolución de los estados $x^{(1)}$ y $x^{(2)}$, respectivamente. La Fig. 3.6c muestra la traza de la matriz P_{k-N}^{-1} utilizada por cada uno de los algoritmos. Podemos ver que la matriz de MHE_{KF} converge a un valor de

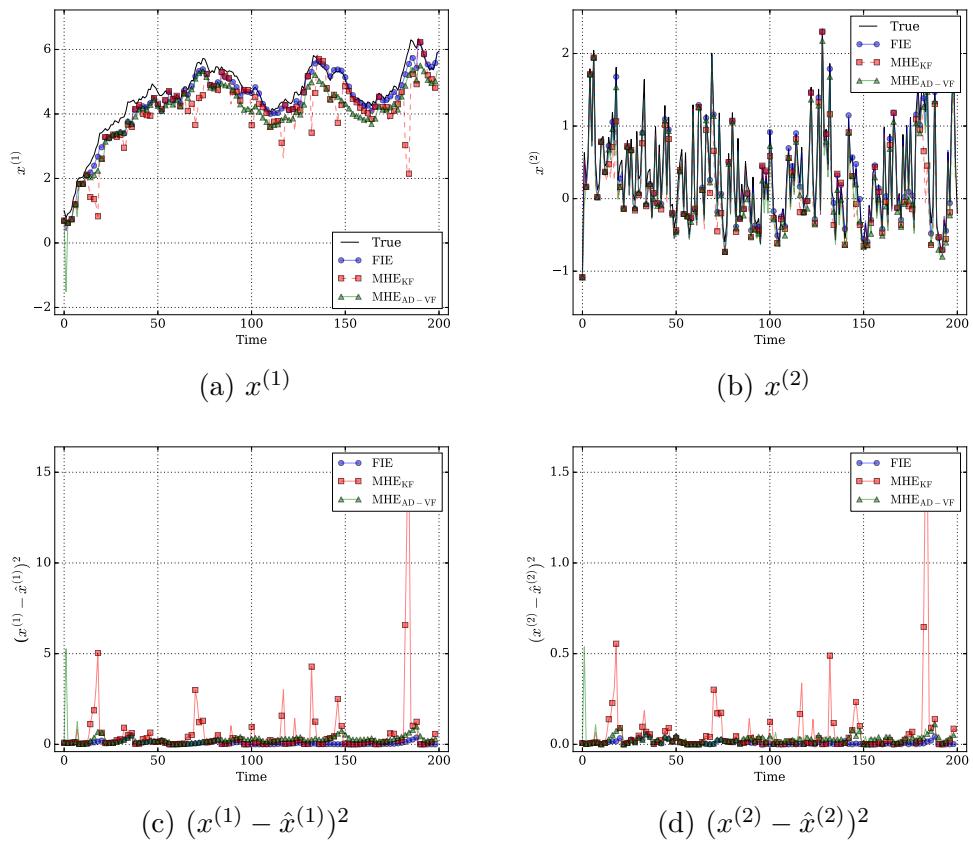


Figura 3.5: Comparación de los estimadores para el modelo (3.19) con horizonte de $N = 5$.

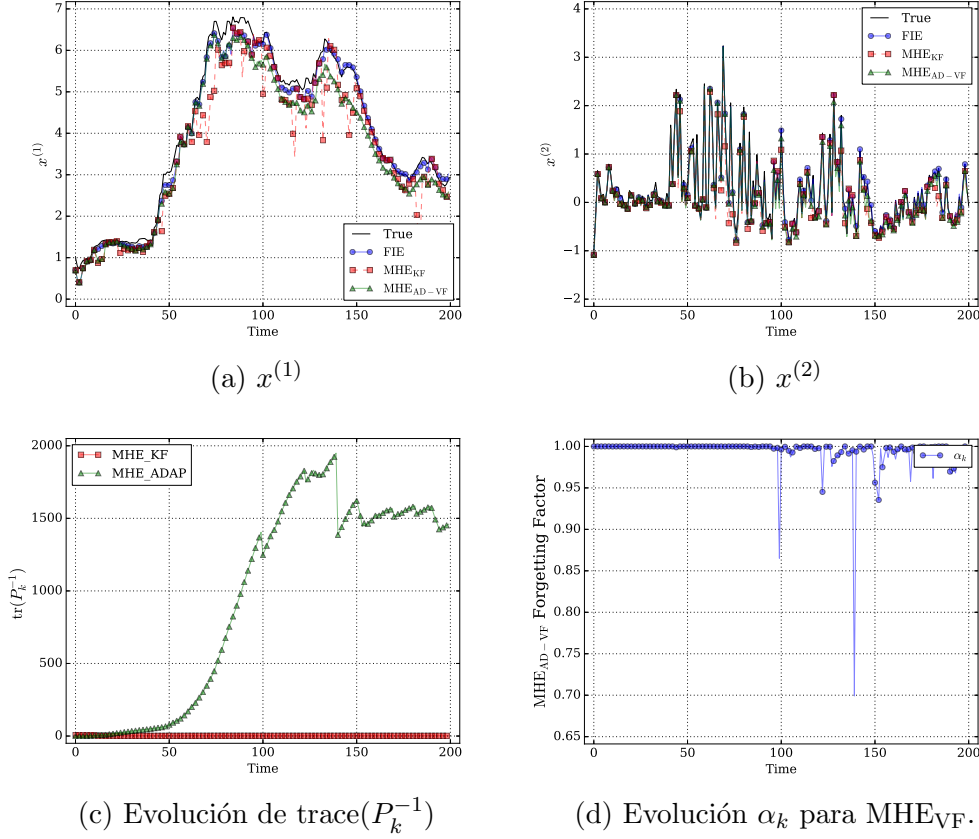


Figura 3.6: Comparación de los estimadores para el modelo (3.19) con horizonte de $N = 5$ y σ_w variable.

estado estacionario y luego no cambia, mientras que el algoritmo propuesto adapta el peso de esta matriz acorde a las condiciones cambiantes del ruido de proceso. En estos casos, la traza de P_{k-N}^{-1} crece cuando el error es pequeño, mostrando que tiene mas confianza en los estimados. Por otro lado, cuando la diferencia entre el estado real y el estimado crece, la traza de la matriz decrece; logrando así mejor adaptación, aún bajo la influencia de perturbaciones desconocidas y cambiantes. En la figura 3.6d se puede ver la evolución del factor de olvido variable de MHE_{VF}. Se puede observar que α_k toma un valor muy cercano a 1 y el algoritmo se comporta como un estimador de mínimos cuadrados ponderado con una gran memoria. En los casos que el ruido crece y el estimador es incapaz de seguir al sistema, el factor de olvido decrece con el fin de permitir una adaptación rápida de la matriz de peso.

3.8. Navegación con MHE

Para poder controlar de forma eficiente un vehículo no tripulado, se debe conocer con la mayor precisión posible la posición, velocidad y orientación. Debe quedar claro que no podemos saber hacia donde dirigirnos si no sabemos donde estamos. Se utilizará MHE para fusionar datos provenientes de un sensor de GPS, acelerómetro, giróscopo y magnetómetro. Para poder fusionar estos datos, es necesario saber en qué marco de referencia se realiza cada una de las mediciones a fin de transformarlas para ser utilizadas en el marco de referencia deseado. Para ello se utilizarán cuatro marcos de referencia:

- ECI: El marco de referencia Earth Centered Inertial (ECI) definido $\{i\} = (x^i, y^i, z^i)$ es un marco de referencia inercial fijo en el espacio. El origen está ubicado en el centro de la Tierra. Se lo define con el eje x apuntando al equinoccio de marzo y el eje z apunta al eje de rotación terrestre. El eje y completa el sistema de coordenadas ortogonal de forma tal que cumpla con la regla de la mano derecha.
- ECEF: El marco de referencia Earth Centered Earth Fixed (ECEF) $\{e\} = (x^e, y^e, z^e)$ tiene su origen en el centro de la Tierra. Se lo define con el eje x apuntando a la intersección de longitud 0° (el meridiano de Greenwich) y latitud 0° (el Ecuador). El eje z apunta al eje de rotación de la Tierra y el eje y completa el sistema de coordenadas ortogonal de forma tal que cumpla con la regla de la mano derecha. El marco ECEF rota junto con la Tierra relativo al marco ECI a una velocidad $\omega_e = 7.2921 \cdot 10^{-5}$ rad/s. En este marco se pueden utilizar coordenadas Cartesianas o elipsoidales (longitud, latitud, altitud) para representar una posición.
- ENU: el sistema East North Up (ENU) $\{n\} = (x^n, y^n, z^n)$ está definido relativo al elipsoide de referencia terrestre (WGS-84). El eje z apunta hacia arriba, con dirección perpendicular al plano tangente del elipsoide, el eje x apunta al este y el eje y al norte, completando así el marco de referencia ortogonal que cumple con la regla de la mano derecha.
- BODY: el marco de referencia fijo al chasis del vehículo $\{b\} = (x^b, y^b, z^b)$ es un marco de referencia móvil que se traslada y rota dado que está fijo al vehículo. El eje x apunta hacia adelante, el eje y apunta hacia la izquierda y el eje z hacia arriba. La posición y orientación del vehículo son descriptas relativas al marco de referencia inercial (aproximado por $\{n\}$ para navegación local).

El sistema de ecuaciones que describe la dinámica de un cuerpo rígido están dadas por

$$\dot{p}^e = v^e + w_p \quad (3.20)$$

$$\dot{v}^e = -2S(\omega_{ie}^e)v^e + a^e + g^e(p^e) + w_v \quad (3.21)$$

$$\dot{q}_b^e = \frac{1}{2}q_b^e \cdot \tilde{\omega}_{ib}^b - \frac{1}{2}\tilde{\omega}_{ie}^e \cdot q_b^e + w_q \quad (3.22)$$

$$\dot{\alpha} = 0 + w_\alpha \quad (3.23)$$

$$\dot{\beta} = 0 + w_\beta \quad (3.24)$$

donde p^e es la posición en coordenadas ECEF, v^e es la velocidad lineal en coordenadas ECEF, a^e es la aceleración lineal en coordenadas ECEF y g^e es el vector gravedad en coordenadas ECEF. El vector gravedad es una función de la posición y es modelado utilizando el modelo gravitatorio J_2 [48]. La velocidad de rotación de la Tierra sobre el eje z es representada por el vector ω_{ie}^e . El cuaternión q_b^e determina la orientación del cuerpo rígido y α y β son los *bias* del giróscopo y acelerómetro, respectivamente.

Las ecuaciones de medición están dadas por

$$\omega_m^b = \omega^b + \alpha + v_\omega \quad (3.25)$$

$$a_m^b = R(q_b^e)a^e + \beta + v_a \quad (3.26)$$

$$m_m^b = R(q_b^e)m^e + v_m \quad (3.27)$$

$$p_m^e = p^e + v_p \quad (3.28)$$

$$v_m^e = v^e + v_v \quad (3.29)$$

donde m^e es la magnitud del campo magnético terrestre en función de la latitud y longitud, w^b y a^e son estados algebraicos y a la vez, variables del problema de optimización.

De esta forma, el vector de estados diferenciales se define como $x = [p^e \ v^e \ q_b^e \ \alpha \ \beta]^T$, el vector de estados algebraicos se define como $z = [\omega^b \ a^e]^T$ y el vector de mediciones se define como $y = [\omega_m^b \ a_m^b \ m_m^b \ p_m^e \ v_m^e]^T$. El índice Ψ a minimizar con respecto a $\hat{x}_{k-N|k}$, $\hat{z}_{k-N|k}$ y \hat{w}_k se define como

$$\Psi_k^N = \|1 - \|\hat{q}_b^e(k-N)\|_2\|_{P_0}^2 + \|\hat{x}_{k-N|k} - \bar{x}_{k-N|k}\|_{P_1}^2 + \quad (3.30)$$

$$\|\hat{z}_{k-N|k} - \bar{z}_{k-N|k}\|_{P_2}^2 + \sum_{j=k-N}^k \|\hat{w}_{j|k}\|_Q^2 + \|\hat{v}_{j|k}\|_R^2. \quad (3.31)$$

El primer término de Ψ en conjunto con las siguientes restricciones

$$\begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \leq \hat{q}_b^e \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \quad (3.32)$$

garantizan la norma unitaria del cuaternión y evitan tener que realizar su normalización fuera del proceso de estimación.

Para las pruebas de fusión de datos se utilizará un vehículo de tracción diferencial configurado en el simulador **Gazebo** (<http://gazebo.org/>) [59]. Este vehículo está dotado de sensores de GPS, acelerómetro, giróscopo y magnetómetro, los cuales proveerán las mediciones para evaluar el desempeño de los algoritmos propuestos.

En esta sección se utilizará MHE en ECEF. Para ello, las coordenadas geodésicas de las mediciones de GPS serán transformadas a ECEF mediante la ecuación

$$\begin{bmatrix} x^e \\ y^e \\ z^e \end{bmatrix} = \begin{bmatrix} (N+h) \cos \phi \cos \lambda \\ (N+h) \cos \phi \sin \lambda \\ (b^2 N/a^2 + h) \sin \phi \end{bmatrix}, \quad (3.33)$$

donde

$$N = \frac{a^2}{\sqrt{a^2 \cos^2 \phi + b^2 \sin^2 \phi}}$$

es el radio de curvatura este-oeste, ϕ es la latitud en radianes, λ la longitud en radianes, $a = 6378137$ m es la longitud del semieje mayor del elipsoide de referencia terrestre y $b = 6356752.3142$ m es la longitud del semieje menor del elipsoide de referencia terrestre.

Para el caso en que el vehículo deba trasladarse distancias grandes, es conveniente que las coordenadas se encuentren en ECEF, mientras que para distancias cortas es conveniente utilizar ENU. Por lo tanto, si se desea utilizar MHE en ENU, las mediciones del GPS pueden transformarse a ENU mediante los siguientes pasos

1. Determinar la latitud, longitud y altitud del punto que determina la posición inicial o de referencia (ϕ_0, λ_0, h_0) y calcular las coordenadas ECEF correspondientes. De esta forma, utilizando la ecuación (3.33) obtenemos p_0^e . Este punto será el origen del marco de coordenadas ENU.
2. Transformar las mediciones de GPS a coordenadas ECEF utilizando la ecuación (3.33) para obtener p^e y calcular los desplazamientos relativos

en ENU mediante

$$p^n = R_n^e(\phi_0, \lambda_0)^T(p^e - p_0^e), \quad (3.34)$$

donde $R_n^e(\phi_0, \lambda_0)$ es la matriz de rotación de ENU a ECEF en función de la latitud ϕ_0 y longitud λ_0 .

La matriz de rotación que transforma de ENU a ECEF está dada por dos rotaciones [49]:

1. Una rotación de $90 - \phi$ en sentido horario sobre el eje E, para alinear el eje U con z^e . Esto es $R_1(-(\pi/2 - \phi))$.
2. Una rotación de $90 + \lambda$ en sentido horario sobre el eje z^e para alinear el eje E con x^e . Esto es $R_3(-(\phi/2 + \lambda))$

por lo que se obtiene

$$\begin{bmatrix} x^e \\ y^e \\ z^e \end{bmatrix} = R_3(-(\phi/2 + \lambda)) R_1(-(\pi/2 - \phi)) \begin{bmatrix} E \\ N \\ U \end{bmatrix}, \quad (3.35)$$

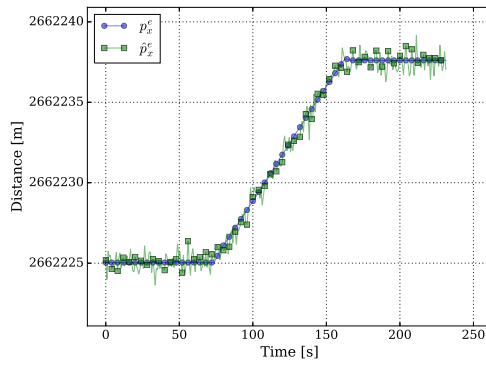
teniendo en cuenta las propiedades de las matrices de rotación, la transformación de ECEF a ENU se realiza mediante la traspuesta de la matriz anterior. Con esta ecuación podemos transformar las mediciones que se realizan en ECEF y obtener su expresión en ENU.

En este caso, se establece un sistemas de coordenadas locales relativas a la posición de referencia. Se debe reemplazar el cuaternión q_b^e por q_b^n . Además, se debe tener en claro en qué marco de referencia se realiza cada una de las mediciones de los sensores y se deberán aplicar las rotaciones necesarias previamente.

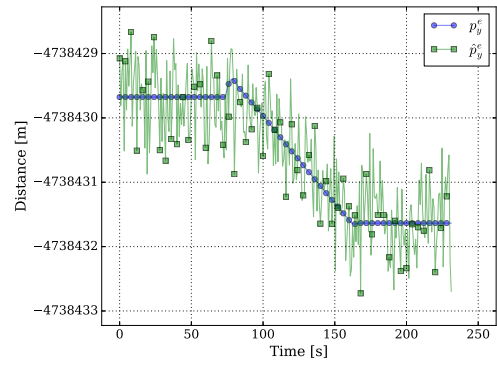
3.9. Resultados

La maniobra realizada por el vehículo consiste en dirigirse desde la posición $(0, 0, 0)$ hacia $(10, 15, 0)$ en el marco de referencia ENU. El simulador utilizado se configura con los parámetros WGS-84 de latitud y longitud de referencia en el estacionamiento de FICH, $(-31.640333^\circ, -60.671112^\circ)$ y los valores del campo magnético se configuran a los correspondiente a la ciudad de Santa Fe, $m^e = (2.701 \times 10^{-6}, 1.82089 \times 10^{-5}, -1.31021 \times 10^{-5})$ ¹. Los ruidos de medición se modelan como ruidos Gaussianos de media cero

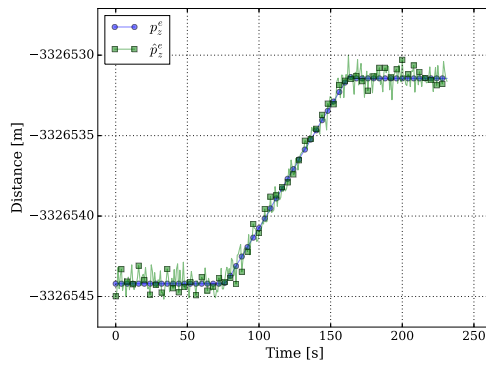
¹Estos datos se obtuvieron de <https://www.ngdc.noaa.gov/geomag-web>.



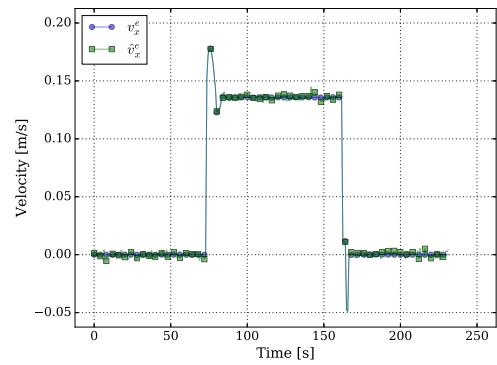
(a) \hat{p}_x^e y p_x^e



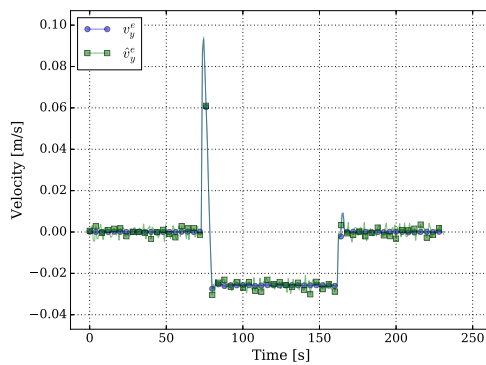
(b) \hat{p}_y^e y p_y^e



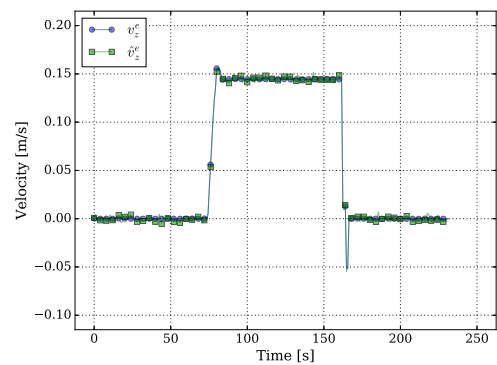
(c) \hat{p}_z^e y p_z^e



(d) \hat{v}_x^e y v_x^e

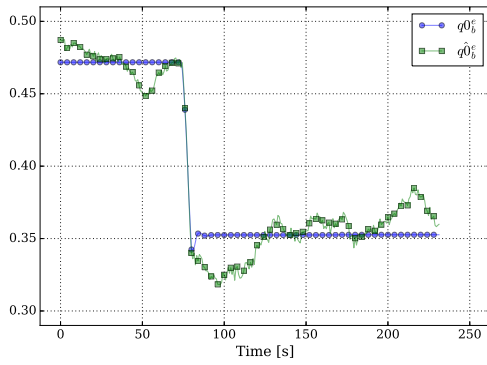


(e) \hat{v}_y^e y v_y^e

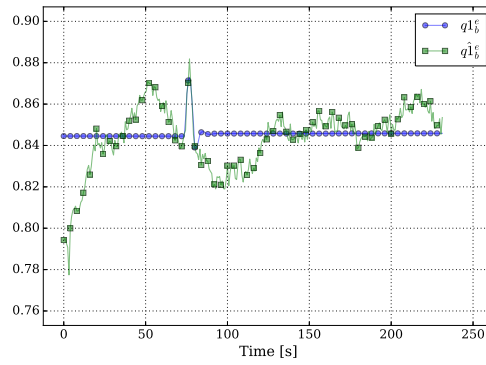


(f) \hat{v}_z^e y v_z^e

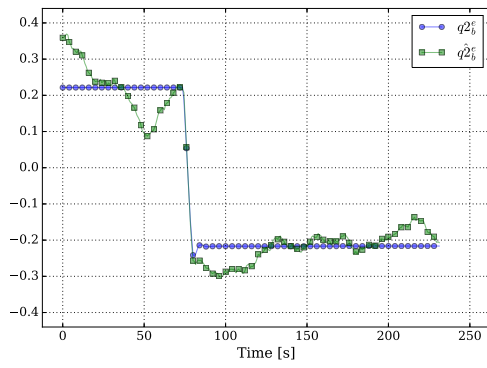
Figura 3.7: Estimación de la posición p^e y la velocidad v^e .



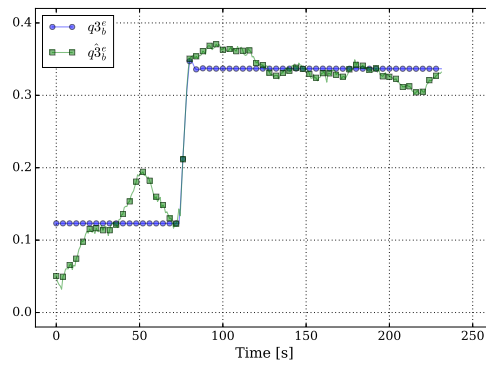
(a) $\hat{q}0_b^e$ y $q0_b^e$



(b) $\hat{q}1_b^e$ y $q1_b^e$

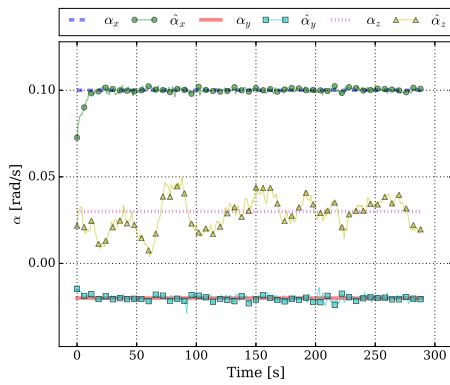


(c) $\hat{q}2_b^e$ y $q2_b^e$

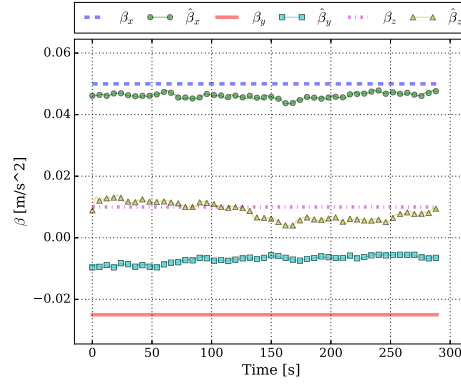


(d) $\hat{q}3_b^e$ y $q3_b^e$

Figura 3.8: Comparación del cuaternión estimado \hat{q}_b^e y el real q_b^e .



(a) $\hat{\alpha}$ y α



(b) $\hat{\beta}$ y β

Figura 3.9: Estimación de los *bias* α y β .

y varianzas definidas por $\sigma_\omega = 1 \times 10^{-4}$, $\sigma_a = 3 \times 10^{-4}$, $\sigma_m = 1 \times 10^{-5}$, $\sigma_p = 5 \times 10^{-1}$ y $\sigma_v = 2 \times 10^{-3}$ para las mediciones de velocidad angular, aceleración lineal, campo magnético, posición y velocidad, respectivamente. Los *bias* correspondientes al giróscopo y acelerómetro se configuran en $\alpha = (0.1, -0.02, 0.03)$ y $\beta = (0.05, -0.025, 0.01)$. El horizonte de estimación es $N = 5$ y las matrices de peso para el problema de MHE se configuran en:

$$\begin{aligned} Q &= 0.001 I_{16 \times 16} \\ R &= \text{diag}([100, 100, 100, 100, 100, 100, \\ &\quad 5000, 5000, 5000, 1, 1, 1, 1, 1, 1]) \\ P_0 &= 0.1 \\ P_1 &= \text{diag}([1, 1, 1, 1, 1, 1, 5, 5, 5, 5, \\ &\quad 10, 10, 10, 10, 10, 10]) \\ P_2 &= I_{6 \times 6} \end{aligned}$$

Las Figuras 3.7, 3.8 y 3.9 muestran los resultados obtenidos de la utilización de MHE para la fusión de datos. En las Figuras 3.7a, 3.7b y 3.7c podemos ver la estimación de la posición en las coordenadas x , y y z en ECEF y su comparación con la posición real proporcionada por el simulador. De la misma forma, en las Figuras 3.7d, 3.7e y 3.7f se puede ver la estimación de la velocidad lineal en las coordenadas x , y y z en ECEF y su comparación con la velocidad lineal real proporcionada por el simulador. Las Figuras 3.8a, 3.8b, 3.8c y 3.8d muestran la estimación de cada una de las componentes del cuaternión y su comparación con la orientación real del vehículo en el marco ECEF. En este aspecto, podemos notar dos particularidades. En primer lugar, si bien el cuaternión facilita el aspecto numérico al no utilizar funciones trigonométricas y evita el *gimball lock*, su interpretación es dificultosa (si no imposible) y esto requiere que en ocasiones, se deba transformarlo a ángulos de Euler. En segundo lugar, al movernos en el marco de referencia ECEF, es más difícil aún tener una idea intuitiva de como estamos posicionados y orientados, ya que nos estamos moviendo sobre una superficie que no es plana y se encuentra rotando sobre el eje z^e , motivo por el cual se debe tener muy claro en qué marco nos movemos y como transformar de forma acorde los estados y mediciones necesarias para evitar incongruencias. Finalmente, las Figuras 3.9a y 3.9b muestran la estimación de los *bias* de giróscopo y acelerómetro. En el caso de α , se puede apreciar una rápida convergencia hacia los valores reales, mientras que para el caso de β estos valores presentan cierto *offset*. El algoritmo utilizado registra tiempos de 50ms para completar cada una de las iteraciones del proceso de estimación, lo que facilita su implementación en vehículos reales, en conjunto con el proceso de control y en sistemas de tiempo real.

3.10. Conclusiones

En este capítulo se investigó la aplicación de un método alternativo para aproximar el costo de arribo en problemas de MHE lineales y no lineales. No se ha encontrado bibliografía que utilice técnicas de estimación adaptativa para estimar la matriz de pesos P_{k-N} en conjunto con MHE. A partir de los resultados obtenidos, podemos afirmar que las técnicas de estimación adaptativa logran buenas aproximaciones del costo de arribo y obtienen buenos resultados, aún cuando se utilizan horizontes de estimación reducidos. La capacidad de adaptación de los factores de olvido variable permiten que el estimador incorpore información relevante a partir de los datos y responda a tanto cambios lentos como a cambios bruscos en la dinámica del sistema.

La aproximación del costo de arribo en problemas no lineales es un tema que es sujeto de estudio. Los resultados aquí presentados son preliminares y no se han estudiado las propiedades de estabilidad o convergencia del estimador no lineal con actualización adaptativa del costo de arribo. Sin embargo, los resultados obtenidos son prometedores, ya que los algoritmos propuestos son capaces de aproximar el costo de arribo utilizando una actualización recursiva de la matriz de pesos de MHE basándose en información obtenida únicamente de los estados del proceso.

Finalmente, se utilizó MHE para resolver el problema de estimación de posición, velocidad y orientación mediante la fusión de datos provenientes de GPS, acelerómetro, giróscopo y magnetómetro. De esta forma, es posible obtener una estimación de la orientación del vehículo en el marco ECEF. Esto nos acerca un paso más a poder controlar vehículos terrestres en terreno arbitrario, teniendo en cuenta no solo la posición obtenida mediante el GPS, sino también su actitud. La solución de este problema requiere un *know-how* muy específico, se debe tener muy en claro en que marco de referencia se realiza cada una de las mediciones y a qué ruido está sujeto cada uno de los sensores y como sintonizar las matrices de peso con el fin de obtener buenos resultados. En caso de que se desee implementar el algoritmo de fusión de datos con MHE, se debe estudiar como manipular los datos provenientes de los distintos sensores a distintas frecuencias de muestreo, sobre todo si se tiene en cuenta que la frecuencia de muestreo de los GPS ronda entre los 1 Hz y 10 Hz, mientras que la frecuencia de muestreo de una IMU ronda los 400 Hz para el caso de comunicación I²C y 1 MHz para el caso de comunicación SPI.

Se debe recalcar que en todos los casos, gracias a la utilización de CasADi [5], los algoritmos desarrollados resuelven en problema de optimización en el orden de los 100ms, lo que permite la operación en tiempo real.

Capítulo 4

Guiado

4.1. Introducción

Las técnicas de control predictivo (MPC, Model Predictive Control) hacen referencia a un conjunto de algoritmos que utilizan modelos del sistema para predecir su comportamiento futuro sobre un horizonte de predicción. Su formulación resuelve un problema de optimización *on-line*, donde la secuencia de entradas de control óptimas es calculada al minimizar una función objetivo sujeta a restricciones. El primer elemento de dicha secuencia es aplicado a la planta siguiendo una estrategia de horizonte móvil [72, 99], tal como se puede apreciar en la Figura 4.1. MPC lineal ha sido aplicado con éxito a una gran variedad de casos debido a su capacidad de incorporar de forma explícita restricciones al modelo del sistema, los estados o las entradas para el cálculo de la secuencia de control [1, 4, 73, 128].

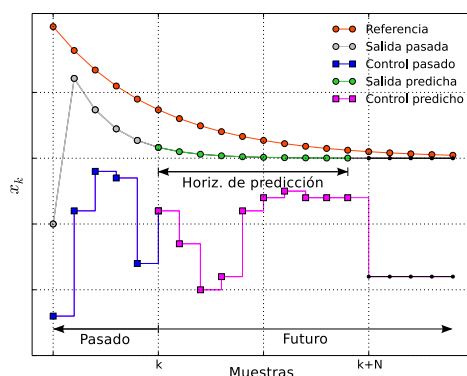


Figura 4.1: Esquema del funcionamiento de MPC.

En las últimas décadas, los principios utilizados por MPC fueron extendidos a sistemas no lineales, lo que condujo al control predictivo no lineal (NLMPC, Non-linear Model Predictive Control). Debido a esto, la utilización de algoritmos de Non-linear Program (NLP) ha sido propuesta en varios trabajos [11, 30, 65, 69]. Sin embargo, los métodos cuyas soluciones se basan en NLP presentan algunas desventajas. Por un lado, son computacionalmente demandantes, ya que requieren resolver un problema de optimización no lineal en tiempo real. Además, las restricciones que impuestas por las características del problema de control pueden conducir a problemas de optimización no convexos.

En varios trabajos [12, 80] se ha adoptado el uso de linealizaciones y aproximaciones lineales para lidiar con el problema de la complejidad computacional. La principal ventaja de estos métodos radica en el hecho que el modelo utilizado para el cálculo de la predicción termina siendo un conjunto de aproximaciones lineales de la planta, lo que convierte el NLP en un conjunto local de problemas lineales convexos [25, 26, 67]. Sin embargo, al resolver el NLMPC utilizando las técnicas de MPC, no se asegura automáticamente la estabilidad a lazo cerrado, tal como describen Morari [80] y Mayne [79] en sus trabajos, donde se presenta una visión general del problema. Una forma de abordar el problema de la estabilidad es agregando una restricción contractiva al problema de optimización. Esta idea fue introducida por Yang y Polak [132]; y la demostración de estabilidad fue realizada por De Oliveira Kothare y Morari [27]. Al utilizar esta metodología, los autores proponen agregar una restricción contractiva que fuerza a los estados del sistema a decrecer en cada instante. Hasta donde sabemos, hay muy pocos trabajos que aborden el agregado de este tipo de restricción contractiva y a la vez, cuando se utilizó, fue para forzar la contracción de los estados del sistema.

En este capítulo se presenta un algoritmo de control robusto para sistemas no lineales. El algoritmo propuesto utiliza un proceso de linealización junto con trayectorias predefinidas que transforman el problema de optimización no convexo en un conjunto local de problemas convexos, los cuales pueden ser resueltos utilizando Quadratic Programming (QP). Para abordar la estabilidad y convergencia, se agrega un conjunto de restricciones contractivas al problema de optimización. Estas restricciones fuerzan que la función de costo decrezca o que permanezca constante en cada instante, permitiendo de esta manera tener en cuenta perturbaciones y determinando una cota superior del valor de la función de costo. Además, se agrega un bucle interno a fin de tener en cuenta errores de linealización y permitiendo así, obtener resultados más precisos.

4.2. Control predictivo no lineal

MPC utiliza un modelo del sistema para predecir su comportamiento. Para el caso de sistemas discretos no lineales, se representan mediante

$$x_{k+1} = f(x_k, u_k, d_k) \quad (4.1)$$

donde $x_k = x(k) \in \mathfrak{R}^{n_x}$, $u_k = u(k) \in \mathcal{U} \subseteq \mathfrak{R}^{n_u}$ y $d_k = d(k) \in \mathcal{D} \subseteq \mathfrak{R}^{n_d}$ son el vector de estados, el vector de entradas de control y el vector de perturbaciones acotadas, respectivamente, \mathcal{U} es el conjunto de restricciones para las entradas de control y $f(\cdot)$ es una función vectorial continua y diferenciable que describe la dinámica del sistema.

El problema de control predictivo basado en modelos no lineales es formulado como un problema de regulación de la siguiente manera:

Dada^[1] una secuencia de perturbaciones

$$\mathbf{d}_k = [d_{k|k}, \dots, d_{k+N-1|k}]^T, \quad (4.2)$$

encontrar a cada instante k , una secuencia de entradas de control

$$\mathbf{u}_k = [u_{k|k}, \dots, u_{k+N-1|k}]^T, \quad (4.3)$$

y secuencia de estados predichos

$$\mathbf{x}_k = [x_{k+1|k}, \dots, x_{k+N|k}]^T, \quad (4.4)$$

sobre un horizonte de predicción de N muestras, tal que sean solución del siguiente problema de optimización

$$\min_{\mathbf{u}_k \in \mathcal{U}} \mathcal{J}(k) = \sum_{i=0}^{N-1} \ell(x_{k+i|k}, u_{k+i|k}) + \ell_N(x_{k+N|k}) \quad (4.5)$$

$$\text{st. } x_{k+1|k} = f(x_{k|k}, u_{k|k}, d_{k|k}).$$

Donde $\ell(\cdot)$ y $\ell_N(\cdot)$ son funciones convexas que representan los costos de etapa y terminal^[2], respectivamente. Los vectores $d_{k+i|k}$, $u_{k+i|k}$ y $x_{k+i|k}$ en (4.2)–(4.4) representan a los vectores de perturbaciones, entradas de control y estados a tiempo $k+i$ que son predichos utilizando la información disponible a tiempo k ^[3] y el modelo del sistema. El óptimo del problema (4.5) será denotada como

$$\mathbf{u}_k^* = [u_{k|k}^*, \dots, u_{k+N-1|k}^*]^T. \quad (4.6)$$

¹Si d_k no está disponible, la suposición generalizada es $d_{k+i} = d_{k+i-1}$, $i = 1, \dots, N$

²Generalmente se utilizan funciones cuadráticas para el costo de etapa y el costo terminal.

³Cuando la referencia al tiempo k sea clara, se hará omisión de dicha información, es decir $(\cdot)_{k+i|k} = (\cdot)_{k+i}$

Independientemente de que $\mathcal{J}(k)$ sea convexa o no, el problema de optimización (4.5) es no-convexo debido a la no-linealidad de la dinámica del sistema y por ello, el esfuerzo computacional es una de las principales consideraciones para su implementación on-line. Si se elige $\mathcal{J}(k)$ de forma tal que sea una función de costo cuadrática, se puede recuperar la convexidad de (4.5) al aproximar el modelo no lineal (4.1) con un modelo lineal variante en el tiempo (LTV, Linear Time Varying) [36, 37], el cual se puede obtener linealizando el sistema alrededor de una trayectoria de estados \mathbf{x}_k^r y entradas de control \mathbf{u}_k^r , donde

$$\begin{aligned}\mathbf{x}_k^r &= \left[x_{k+1|k}^r, \dots, x_{k+N|k}^r \right]^T, \\ \mathbf{u}_k^r &= \left[u_{k|k}^r, \dots, u_{k+N-1|k}^r \right]^T.\end{aligned}\tag{4.7}$$

Suponiendo que una secuencia de perturbaciones de referencia $\mathbf{d}_{k+i|k}^r$, con $i = 0, \dots, N-1$ está dada o es estimada, el comportamiento dinámico de la desviación de la trayectoria puede ser escrito como un modelo LTV

$$\tilde{x}_{k+1|k} = A_{k|k} \tilde{x}_{k|k} + B_{u_{k|k}} \tilde{u}_{k|k} + B_{d_{k|k}} \tilde{d}_{k|k},\tag{4.8}$$

donde

$$\tilde{x}_{k|k} = x_{k|k} - x_{k|k}^r, \tilde{u}_{k|k} = u_{k|k} - u_{k|k}^r, \tilde{d}_{k|k} = d_{k|k} - d_{k|k}^r.\tag{4.9}$$

Las matrices $A_{k|k}$, $B_{u_{k|k}}$ y $B_{d_{k|k}}$ son las matrices de los Jacobianos del sistema (4.1) y se definen como

$$A_{k|k} = \left. \frac{\partial f(x_k, u_k, d_k)}{\partial x_k} \right|_{(*)}, B_{u_{k|k}} = \left. \frac{\partial f(x_k, u_k, d_k)}{\partial u_k} \right|_{(*)}, B_{d_{k|k}} = \left. \frac{\partial f(x_k, u_k, d_k)}{\partial d_k} \right|_{(*)},\tag{4.10}$$

donde $(*)$ determina el triplete de parámetros (x_k^r, u_k^r, d_k^r) . En términos del sistema LTV (4.8), se adopta la siguiente función objetivo cuadrática

$$\mathcal{J}(k) = \sum_{i=0}^{N-1} \|\tilde{x}_{k+i|k}\|_Q^2 + \|\tilde{u}_{k+i|k}\|_R^2 + \|\tilde{x}_{k+N|k}\|_{P_{k|k}}^2,\tag{4.11}$$

donde $Q, R, P_{k|k}$ son matrices positivas definidas; $P_{k|k}$ es la matriz de peso terminal que se elige de manera tal que satisfaga la ecuación de Lyapunov

$$P_{k|k} - A_{k|k}^T P_{k|k} A_{k|k} = Q.\tag{4.12}$$

Como resultado, el problema de optimización no-convexo (4.5) puede ser aproximado como un problema de optimización convexo de la siguiente ma-

nera:

$$\begin{aligned} & \min_{\tilde{\mathbf{u}}_k \in \mathcal{U}} \mathcal{J}(k) \\ \text{st. } & \begin{cases} \tilde{x}_{k+1|k} = A_{k|k}\tilde{x}_{k|k} + B_{u_{k|k}}\tilde{u}_{k|k} + B_{d_{k|k}}\tilde{d}_{k|k}, \\ \tilde{x}_{k|k} = x_{k|k} - x_{k|k}^r, \\ \tilde{u}_{k|k} = u_{k|k} - u_{k|k}^r, \\ \tilde{d}_{k|k} = d_{k|k} - d_{k|k}^r. \end{cases} \end{aligned} \quad (4.13)$$

El algoritmo 2 resume la técnica de NMPC de horizonte móvil.

Algoritmo 2: NMPC de horizonte móvil

Dados $Q, R > 0$, la condición inicial $x_{k|k}$.

Paso 1: Obtener la trayectoria de linealización $\mathbf{x}_k^r, \mathbf{u}_k^r$ utilizando como condición inicial $\mathbf{u}_k^0 = [u_{k|k-1}^*, u_{k+1|k-1}^*, \dots, u_{k+N-2|k-1}^*, 0]^T$ y estimados d_{k+i} para $i = 0, \dots, N-1$

Paso 2: Obtener el sistema LTV (4.8) y $P_{k|k}$ mediante la solución de (4.12)

Paso 3: Calcular la secuencia de entradas de control óptima $\tilde{\mathbf{u}}_k^*$ al resolver (4.13)

Paso 4: Actualizar $\mathbf{u}_k^* \leftarrow \mathbf{u}_k^r + \tilde{\mathbf{u}}_k^*$

Paso 5: Aplicar $u_{k|k} = u_{k|k}^*$ al sistema

Paso 6: Mover el horizonte al siguiente instante de muestreo $k \leftarrow k+1$ y volver al **Paso 1**

Las técnicas de linealización son la manera más sencilla para adaptar métodos de control lineal a problemas de control no lineal. En ausencia de perturbaciones y errores de linealización, el Algoritmo 2 garantizará la estabilidad a lazo cerrado.

Suposición 4.1. El sistema LTV (4.8) es estabilizable para $\mathbf{u}_k \in \mathcal{U}$.

Suposición 4.2. El horizonte de predicción N es lo suficientemente largo.

Suposición 4.3. No hay perturbaciones, es decir $d_{k+i} = 0, i = 0, \dots, N-1$.

Teorema 4.1. Sean las suposiciones 4.1–4.3 válidas. Si el problema de optimización (4.13) resuelto utilizando el Algoritmo 2 es factible, entonces el origen es un punto de equilibrio exponencialmente estable.

Demostración. Ver Apéndice B.1. □

A pesar de que la Suposición [4.2](#) establece que el horizonte de predicción N debe ser lo suficientemente largo, para aplicaciones de ingeniería este horizonte debe elegirse lo más corto posible a fin de que reduzca la carga computacional. En consecuencia, la estabilidad del sistema debe garantizarse utilizando un argumento diferente (ver, por ejemplo, [\[26, 27, 79\]](#)). Es más, el Teorema [4.1](#) puede no satisfacerse en presencia de perturbaciones debido a que no se puede garantizar la contractividad de la función de costo en instantes sucesivos. Para hacer frente a este problema, se propone agregar la restricción contractiva convexa

$$\mathcal{J}(k) \leq \mathcal{J}_0(k), \quad (4.14)$$

al problema de optimización [\(4.13\)](#), donde $\mathcal{J}_0(k)$ denota la función de costo evaluada en la solución inicial

$$\mathbf{u}_k^0 = [u_{k|k-1}^*, u_{k|k-1}^*, \dots, u_{k+N-2|k-1}^*, 0]^T. \quad (4.15)$$

en la iteración k . Nótese que esta restricción fuerza que la función de costo se mantenga constante o decrezca en el instante actual, determinando de esta manera una cota superior para $\mathcal{J}(k)$. Entonces, el nuevo problema de optimización puede escribirse como:

$$\begin{aligned} & \min_{\tilde{\mathbf{u}}_k \in \mathcal{U}} \mathcal{J}(k) \\ \text{st.} \quad & \begin{cases} \tilde{x}_{k+1|k} = A_{k|k} \tilde{x}_{k|k} + B_{k|k} \tilde{u}_{k|k}, \\ \tilde{x}_{k|k} = x_{k|k} - x_{k|k}^r, \\ \tilde{u}_{k|k} = u_{k|k} - u_{k|k}^r, \\ \mathcal{J}(k) \leq \mathcal{J}_0(k). \end{cases} \end{aligned} \quad (4.16)$$

Debido a que la restricción contractiva [\(4.14\)](#) está definida en el instante actual, si ocurre alguna perturbación el valor de $\mathcal{J}(k)$ puede incrementarse (sólo a tiempo k); pero entonces la restricción contractiva lo fuerza a decrecer o permanecer constante. El problema de optimización [\(4.16\)](#) puede ser visto como un problema multiobjetivo, donde la restricción [\(4.14\)](#) se utiliza para garantizar la estabilidad del sistema a lazo cerrado y $\mathcal{J}(k)$ se utiliza para medir el desempeño del sistema a lazo cerrado.

Teorema 4.2. *Si el problema de optimización [\(4.16\)](#) resuelto utilizando el Algoritmo [2](#) es factible, entonces el sistema a lazo cerrado es estable.*

Demostración. La adición de la restricción [\(4.14\)](#) no afecta la factibilidad o la solución original del problema [\(4.13\)](#). Por este motivo, en ausencia de

perturbaciones la restricción (4.14) garantiza la contractividad de la función de costo a instantes sucesivos, es decir

$$\mathcal{J}^*(k) \leq \mathcal{J}(k) \leq \mathcal{J}_0(k) \leq \mathcal{J}^*(k-1) \leq \mathcal{J}(k-1). \quad (4.17)$$

□

Observación 4.1. Como la estabilidad del sistema está garantizada, se puede reducir el horizonte de predicción N , disminuyendo así la carga computacional para la implementación on-line.

Observación 4.2. La adición de la restricción (4.14) es equivalente a la adición de una restricción en la entrada \mathbf{u}_k , por lo tanto, si el sistema es estabilizable con $\mathbf{u}_k \in \mathcal{U}$, entonces la factibilidad inicial está garantizada y mediante la utilización del argumento de factibilidad recursiva, la restricción contractiva (4.14) no afecta la factibilidad original [79].

4.3. Control predictivo no lineal robusto

El diseño de algoritmos de control robusto ha sido estudiado por muchos años, principalmente porque éstos algoritmos tienen la capacidad para incorporar incertidumbres paramétricas y estructurales del sistema (modeladas como perturbaciones acotadas) durante operación. Una manera de considerar la robustez en algoritmos NLMPC consiste en evaluar a cada instante de muestreo todas las trayectorias posibles del sistema para una perturbación dada (o estimada). Esto puede realizarse resolviendo un problema de optimización que considera las diferentes trayectorias de los estados, es decir

$$\begin{aligned} & \min_{\tilde{\mathbf{u}}_k^l \in \mathcal{U}} \mathcal{J}(k) \\ \text{st.} \quad & \begin{cases} \tilde{x}_{k+1|k}^l = A_{k|k}^l \tilde{x}_{k|k}^l + B_{k|k}^l \tilde{u}_{k|k}^l, \\ \tilde{x}_{k|k}^l = x_{k|k}^l - x_{k|k}^r, \\ \tilde{u}_{k|k}^l = u_{k|k}^l - u_{k|k}^r, \\ \mathcal{J}^l(k) \leq \mathcal{J}_0^l(k), \end{cases} \end{aligned} \quad (4.18)$$

donde $l = 1, \dots, m$ denota las diferentes realizaciones del sistema respecto a la perturbación dada. Como resultado, cada trayectoria definida por los estados puede interpretarse como el borde de un polígono variante en el tiempo [24, 63]. Este polígono puede ser utilizado para generar un tubo que contenga todas las trayectorias posibles de los estados. Los tubos han sido ampliamente utilizados para acotar la incertidumbre [14, 63, 78, 133]. Sin

embargo, determinar un tubo exacto para sistemas no lineales es una tarea muy complicada.

En este trabajo, el sistema LTV (4.8) se obtiene mediante una expansión de series de Taylor de primer orden. Para medir la desviación entre el sistema LTV y el no lineal, se utiliza el resto de Taylor de segundo orden para acotar los errores de linealización. En vez de obtener la secuencia de todas las trayectorias de los estados \mathbf{x}_k^l con $l = 1, \dots, m$, proponemos utilizar el resto de Taylor para calcular la secuencia de trayectorias con la peor incertidumbre \mathbf{x}_k^Δ . Esta trayectoria puede ser utilizada para determinar un tubo que acota a todas las trayectorias de los estados. Finalmente, este tubo es utilizado para garantizar la estabilidad del sistema a lazo cerrado.

El sistema no lineal (4.1) puede ser aproximado exactamente con un modelo LTV si el resto de Taylor de segundo orden $R_1(\tilde{x}_k, \tilde{u}_k, \tilde{d}_k)$ ⁴ se suma al lado derecho de la Ecuación (4.8)

$$\tilde{x}_{k+1|k} = A_{k|k}\tilde{x}_{k|k} + B_{k|k}\tilde{u}_{k|k} + R_1(\tilde{x}_k, \tilde{u}_k, \tilde{d}_k). \quad (4.19)$$

De la ecuación (4.19) se puede ver que el término $R_1(\cdot)$ actúa como una perturbación aditiva. Este término puede ser maximizado⁵ a fin de obtener \mathbf{x}_k^Δ , que es la secuencia de trayectorias de los estados con la peor incertidumbre. Una vez que se obtiene \mathbf{x}_k^Δ , se puede calcular su costo asociado $\mathcal{J}^\Delta(k)$. Entonces, la condición de estabilidad para el problema se puede establecer si se fuerza a la función a decrecer. Esto puede realizarse agregando la siguiente restricción contractiva

$$\mathcal{J}^\Delta(k) \leq \mathcal{J}_0^\Delta(k), \quad (4.20)$$

al problema de optimización (4.16). Finalmente, el problema de control robusto a resolver es

$$\begin{aligned} & \min_{\tilde{\mathbf{u}}_k \in \mathcal{U}} \mathcal{J}(k) \\ \text{st. } & \begin{cases} \tilde{x}_{k+1|k} = A_{k|k}\tilde{x}_{k|k} + B_{k|k}\tilde{u}_{k|k}, \\ \tilde{x}_{k|k} = x_{k|k} - x_{k|k}^r, \\ \tilde{u}_{k|k} = u_{k|k} - u_{k|k}^r, \\ \mathcal{J}(k) \leq \mathcal{J}_0(k), \\ \mathcal{J}^\Delta(k) \leq \mathcal{J}_0^\Delta(k), \end{cases} \end{aligned} \quad (4.21)$$

donde $\mathcal{J}_0^\Delta(k)$ denota la función de costo $\mathcal{J}^\Delta(k)$ evaluada para la condición inicial \mathbf{u}_k^0 .

⁴ $R_1(\cdot)$ puede obtenerse como en [77]

⁵Si no hay información sobre \mathbf{d}_k puede suponer que tenemos un valor dado

Al incluir la restricción contractiva (4.20) se garantiza la estabilidad en el origen ya que $\mathcal{J}^\Delta(k)$ se ve forzada a decrecer (o permanecer constante). Es más, mientras $\mathcal{J}^\Delta(k)$ sea *empujada* a cero, va a contraer el costo nominal. Si se utilizan argumentos similares a los utilizados en la demostración del Teorema 4.2 y en (4.17), se puede demostrar que si el problema de optimización (4.21) resuelto utilizando el Algoritmo 2 es factible, entonces el origen es un punto de equilibrio exponencialmente estable.

Observación 4.3. Nótese que $\mathcal{J}_0^\Delta(k)$ es una cota superior *relajada* y por lo tanto, es más grande que $\mathcal{J}^\Delta(k)$, lo que no afecta la factibilidad del problema (4.21).

Observación 4.4. Siguiendo los mismos argumentos que los utilizados en la Observación 4.2, se puede deducir que agregar la restricción contractiva (4.20) al problema de optimización (4.21) no afecta la factibilidad original.

Vale la pena comparar el enfoque propuesto con los descritos en [14] y [63]. En este trabajo se obtiene un tubo externo simplemente maximizando $R_1(\tilde{x}_k, \tilde{u}_k, \tilde{d}_k)$ y luego calculando la trayectoria de estados \mathbf{x}_k^Δ . Dentro de este tubo se encuentran todas las trayectorias perturbadas del sistema [78, 94]. Además, se garantiza la condición de estabilidad mediante la inclusión de la restricción (4.20). El procedimiento propuesto por Cannon et al. [14] es más complicado. Los autores resuelven un problema de optimización multiparamétrico con muchas restricciones, lo que conduce a una alta carga computacional (aún para el caso de modelos simples) y por este motivo, es imposible utilizar el algoritmo en tiempo real. Por otro lado, Langson et al. [63] proponen un método de MPC robusto para sistemas lineales con restricciones e incertidumbre. Utilizan polítopos compactos convexos y poliedros cerrados convexos, los cuales son complicados de manipular cuando hay varias regiones resultantes. Es más, el tubo se define como una secuencia de conjuntos de estados y su ley de control (variante en el tiempo) asociada. Esto también es muy demandante computacionalmente, ya que calculan todas las trayectorias posibles para definir el tubo.

4.4. Control predictivo no lineal robusto iterado

Cuando se linealizan sistemas no lineales pueden aparecer errores de linealización, los cuales pueden ser grandes si las trayectorias están lejos del punto operativo del sistema. Para tener en cuenta estos errores, proponemos incluir una técnica iterativa [23, 26] en el Algoritmo 2 a fin de mejorar el

desempeño del sistema a lazo cerrado. El mecanismo iterativo propuesto trabaja de la siguiente manera: a cada instante de muestreo, el sistema no lineal es linealizado sobre una trayectoria predefinida. A partir de esta trayectoria, se calcula la secuencia de entradas de control óptimas y se comprueba si se cumple la condición para salir del bucle. Si no se cumple, la trayectoria de linealización es recalculada utilizando la nueva secuencia de control. El sistema no lineal es linealizado nuevamente y la secuencia de entradas de control recalculada. Este bucle continúa hasta que se satisfaga la condición de convergencia. Como resultado, se obtiene una secuencia de entradas de control \mathbf{u}_k^* más precisa. El Algoritmo 3 resume la técnica detallada.

Algoritmo 3: NMPC Robusto Iterado

Dadas $Q, R > 0$, la condición inicial $x_{k|k}$, el índice de iteración q .

Paso 1: Inicializar $\mathbf{u}_k^q = [u_{k|k-1}^*, u_{k+1|k-1}^*, \dots, u_{k+N-2|k-1}^*, 0]^T$

Paso 2: Obtener la trayectoria de linealización $\mathbf{x}_k^q, \mathbf{u}_k^q$

Paso 3: Obtener el sistema LTV (4.8) y $P_{k|k}^q$ resolviendo (4.12)

Paso 4: Calcular la secuencia de entradas de control óptima $\tilde{\mathbf{u}}_k^{*,q}$ resolviendo (4.21)

Paso 5: Actualizar $\mathbf{u}_k^{*,q} \leftarrow \mathbf{u}_k^q + \tilde{\mathbf{u}}_k^{*,q}$

Paso 6: *if* $\left\| \mathbf{u}_k^{*,q} - \mathbf{u}_k^{*,q-1} \right\|_\infty \leq \epsilon$

$$\mathbf{u}_k^* \leftarrow \mathbf{u}_k^{*,q},$$

$$k \leftarrow k + 1$$

$$q \leftarrow 0$$

else

$$q \leftarrow q + 1$$

$$\text{Actualizar } \mathbf{u}_k^q = \mathbf{u}_k^{*,q-1}$$

Volver al **Paso 2**

end

Paso 7: Aplicar $u_{k|k} = u_{k|k}^*$ al sistema y volver al **Paso 1**

Dado que el problema de optimización a resolver en el Algoritmo 3 incluye las restricciones contractivas (4.14) y (4.20), la estabilidad está garantizada. En consecuencia, el proceso iterativo puede detenerse en cualquier instante deseado, disminuyendo así la carga computacional.

Teorema 4.3. *El bucle de iteración del Algoritmo 3 converge al óptimo.*

Demostración. Ver Apéndice B.2. □

4.5. Resultados

En esta sección se muestran ejemplos en simulación. Se utilizará el simulador **Gazebo**, el modelo de un robot de tracción diferencial y la técnica de NLMPC robusto iterativo de la Sección 4.4. Se realizará la maniobra autónoma de ir desde el punto inicial $(0, 0, 0)$ al punto final $(10, 15, 0)$ en el marco de referencia ENU. El estado inicial del controlador es obtenido utilizando MHE para fusionar datos de mediciones de posicionamiento, GPS, acelerómetro, giróscopo y magnetómetro provistas por el entorno de simulación con un período de muestreo de 0.1 s. Se utiliza MHE tal como se describió en la Sección 3.8, con los siguientes parámetros:

$$\begin{aligned}
 Q_{MHE} &= \text{diag}([500, 500, 500, \\
 &\quad 0.1, 0.1, 0.1, \\
 &\quad 0.5, 0.5, 0.5, 0.5, \\
 &\quad 100, 100, 100, \\
 &\quad 50, 50, 50,]) \\
 R_{MHE} &= \text{diag}([5000, 5000, 5000, \\
 &\quad 20000, 20000, 20000, \\
 &\quad 5000, 5000, 5000, \\
 &\quad 1, 1, 1, \\
 &\quad 10, 10, 10]) \\
 P_0 &= 0.1 \\
 P_1 &= \text{diag}([10, 10, 10, \\
 &\quad 1, 1, 1, \\
 &\quad 10, 10, 10, 10, \\
 &\quad 10, 10, 10, \\
 &\quad 10, 10, 10]) \\
 P_2 &= 0.5 I_{6 \times 6}
 \end{aligned}$$

Los ruidos de medición se modelan como ruidos Gaussianos de media cero y varianzas definidas por $\sigma_\omega = 2 \times 10^{-4}$, $\sigma_a = 3 \times 10^{-4}$, $\sigma_m = 1 \times 10^{-5}$, $\sigma_p = 5 \times 10^{-1}$ y $\sigma_v = 5 \times 10^{-3}$ para las mediciones de velocidad angular, aceleración lineal, campo magnético, posición y velocidad, respectivamente. Los *bias* correspondientes al giróscopo y acelerómetro se configuran en $\alpha = (1 \times 10^{-5}, -2 \times 10^{-5}, 3 \times 10^{-5})$ y $\beta = (5 \times 10^{-2}, -2.5 \times 10^{-2}, 1 \times 10^{-2})$. El

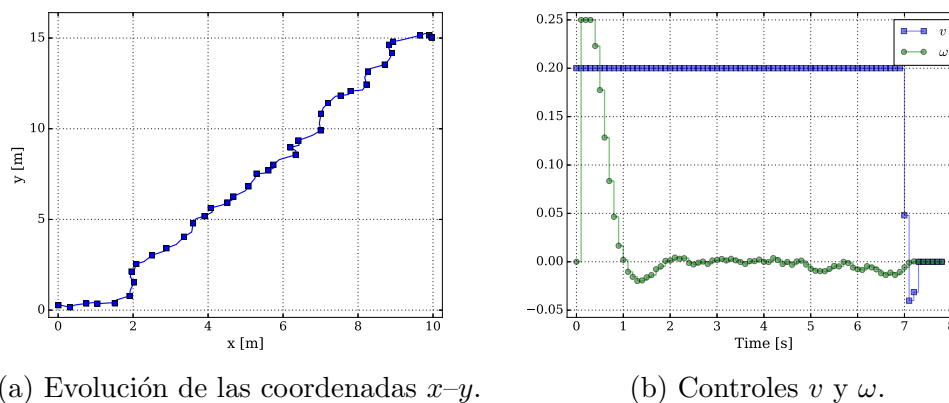
(a) Evolución de las coordenadas x - y .(b) Controles v y ω .

Figura 4.2: Posición y controles aplicados en ENU.

controlador utiliza el siguiente modelo no lineal

$$\begin{aligned}\dot{x} &= v \cos \psi \\ \dot{y} &= v \sin \psi \\ \dot{\psi} &= \omega\end{aligned}\tag{4.22}$$

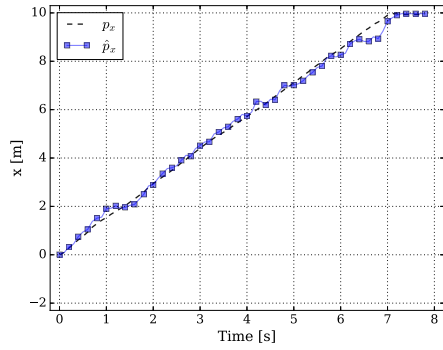
donde el vector de estados (x, y, ψ) representa la posición y orientación del vehículo en el plano x - y y (v, ω) es el vector de acciones de control, el cual representa la velocidad lineal y angular del vehículo, respectivamente. Las matrices de peso se eligen como

$$Q = \text{diag}(25, 25, 0.00001)\tag{4.23}$$

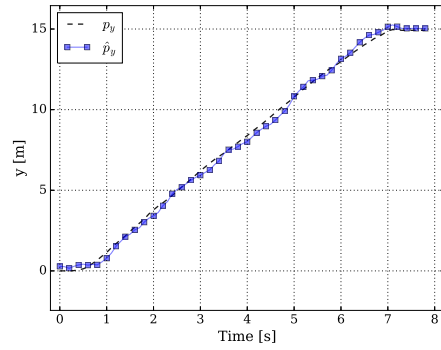
$$Q_n = \text{diag}(100, 100, 0.00001)\tag{4.24}$$

$$R = \text{diag}(30, 50)\tag{4.25}$$

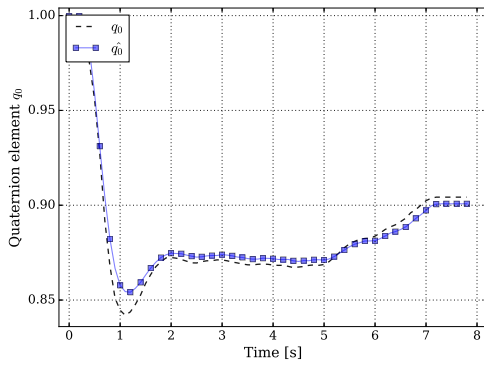
En las Figuras [4.2a](#) y [4.2b](#) se pueden observar la evolución de las coordenadas x - y estimadas con MHE y las acciones de control obtenidas. Se puede apreciar que inicialmente el vehículo gira para alinearse con el punto objetivo para luego dirigirse casi en línea recta hacia el mismo. La Figura [3.7](#) muestra la comparación de los estados estimados de interés y los estados reales obtenidos del simulador utilizado. En todos los casos se puede observar que el algoritmo MHE es capaz de obtener buenas estimaciones de los estados a pesar de poseer mediciones con ruido (en el caso de la posición) o de tener mediciones indirectas (en el caso del cuaternión de orientación), lo que permite un buen desempeño del controlador. Finalmente, la Figura [4.4](#) muestra el tiempo de ejecución utilizado por los algoritmos de MHE y MPC utilizando una notebook con procesador Intel Core i7-5600U de 2.60



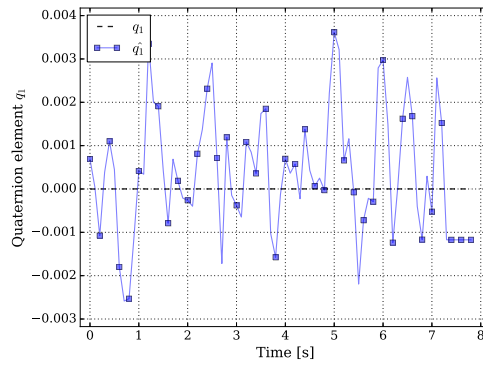
(a) \hat{p}_x y p_x .



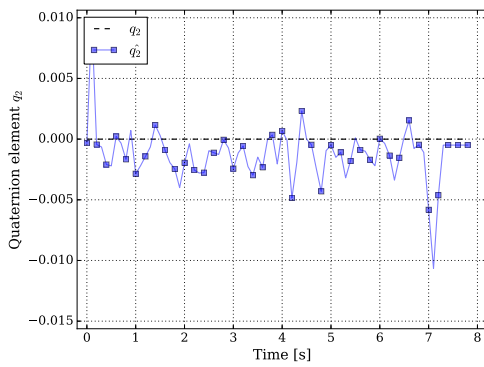
(b) \hat{p}_y y p_y .



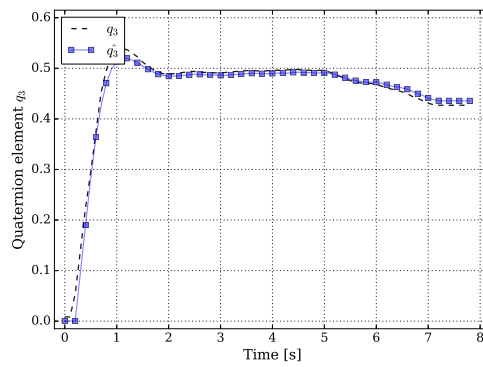
(c) \hat{q}_0 y q_0 .



(d) \hat{q}_1 y q_1 .



(e) \hat{q}_2 y q_2 .



(f) \hat{q}_3 y q_3 .

Figura 4.3: Estimación de la posición p y el cuaternión q en ENU.

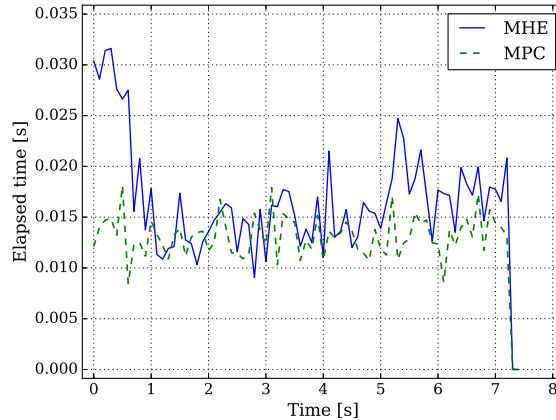


Figura 4.4: Tiempo de ejecución para MHE y MPC con horizonte $N = 5$.

GHz. Como se puede apreciar, en ninguna ocasión se superaron los 50 ms, por lo que la implementación de estimadores y controladores de horizonte móvil utilizando CasADi resulta prometedora en cuanto a la posibilidad de implementación en sistemas que requieran operación en tiempo real.

4.6. Conclusiones

En este capítulo se presentó una técnica de control predictivo no lineal robusta. La técnica propuesta está basada en la linealización de sistemas no lineales a lo largo de una trayectoria predefinida y la minimización de una función objetivo sujeta a restricciones. Para garantizar la estabilidad del sistema a lazo cerrado, se ha agregado una restricción contractiva al problema de optimización. Esta restricción contractiva fuerza que la función de costo decrezca (o permanezca constante) en el instante actual. Dado que la estabilidad del sistema está garantizada (aún ante la presencia de incertidumbres), la inclusión de esta restricción permite reducir el horizonte de predicción a un valor mínimo, reduciendo así la carga computacional. Esto resulta útil cuando se necesita controlar sistemas no lineales de dinámica rápida, tales como vehículos autónomos. La robustez del algoritmo NLMPC propuesto se logra utilizando el residuo de Taylor para calcular la trayectoria de estados asociada a la peor incertidumbre. Esta trayectoria puede ser utilizada para determinar un tubo que contiene a todas las trayectorias posibles del sistema. La metodología propuesta para obtener el tubo para la trayectoria de los estados parece ser más simple y de menor demanda computacional. Para

tener en cuenta los errores de linealización y mejorar el desempeño del sistema a lazo cerrado, se ha incluido un bucle iterativo en el algoritmo NLMPC robusto, lo que conduce al algoritmo NLMPC robusto iterado.

El algoritmo NLMPC iterado fue utilizado como una unidad central para controlar el modelo de un robot de tracción diferencial. Para evaluar el desempeño del algoritmo, se realizó la simulación de la maniobra típica de dirigirse a un punto utilizando como entorno de trabajo el simulador **Gazebo**. El estado inicial del controlador es estimado en cada instante utilizando MHE para fusionar datos de posición, acelerómetro, giróscopo y magnetómetro. Los algoritmos de estimación y control fueron capaces de controlar el vehículo de forma exitosa con tiempos de ejecución menores a los 100 ms en conjunto.

Los tópicos estudiados en estas secciones abren la puerta a la estimación y control de vehículos autónomos que se desplazan por terreno arbitrario utilizando fusión de datos para estimar sus estados en los 6 DOF. Se debe tener en cuenta que este área requiere mucho *expertise* y se deben tener muy afianzados varios de los temas aquí expuestos. La navegación y guiado en ECEF es un tema un poco más complejo, ya que al movernos sobre este último marco de referencia, los estados y los *setpoints* no son tan sencillos de imaginarse, motivo por el cual, se eligió realizar estas simulaciones en el marco de referencia ENU.

Capítulo 5

Múltiples vehículos

5.1. Introducción

Dos de los paradigmas más comunes para resolver problemas de MPC son las estrategias centralizadas y descentralizadas. Las estrategias centralizadas surgen del deseo de operar el sistema de manera óptima, mientras que las estructuras descentralizadas resultan del despliegue gradual del desarrollo de un sistema. Un controlador MPC centralizado puede ser difícil –si no imposible– de implementar en sistemas de gran escala [61, 71]. En las estrategias descentralizadas, el problema MPC que abarca todo el sistema es dividido en subproblemas aprovechando la estructura del sistema y luego, estos subproblemas se resuelve de forma independiente. En general, los esquemas descentralizados aproximan la interacción entre los subsistemas y tratan a las entradas de los otros subsistemas como perturbaciones externas. Esta suposición conduce a un desempeño pobre del sistema [107, 115]. Por este motivo, hay necesidad de una integración funcional entre los controladores descentralizados [2, 7, 16, 17, 135].

Existen muy pocas formulaciones de MPC distribuido en la literatura. En [35] se propuso un esquema MPC distribuido para la clase de sistemas cuyos subsistemas tienen dinámicas independientes, pero están vinculadas a través de sus funciones de costo y restricciones. Dunbar [34] extendió el método descrito en [35] y da soporte a sistemas cuyos subsistemas tienen dinámicas débilmente vinculadas. El desempeño de este esquema MPC distribuido es diferente del centralizado en la mayoría de los casos [35]. Se han propuesto algoritmos de MPC distribuidos para sistemas lineal invariante en el tiempo (LTI, Linear Time Invariant) sin restricciones en [13, 52, 134] y [125]. En [13] y [52], se supone que la evolución de los estados de cada subsistema sólo está influenciada por los estados de los subsistemas que interactúan y

por las entradas de los subsistemas locales; mientras que esta restricción fue removida en [125] y [134]. Esta elección del modelo restringe los sistemas en que se puede aplicar el algoritmo, debido a que en muchos casos la evolución de los estados del subsistema está influenciada por las entradas de los sistemas interconectados. En estos esquemas, es aún más crítico el hecho que los MPCs de cada subsistema solo conocen la función de costo y restricciones del propio subsistema. Un esquema MPC distribuido donde el efecto de cada subsistema interconectado es tratado como incertidumbres acotadas se propuso en [53]. Sin embargo, no se establecieron condiciones de estabilidad y optimalidad; y tampoco se estudió el efecto de fallas en la comunicación entre subproblemas.

En este capítulo abordamos el problema de MPC distribuido visto desde una perspectiva de teoría de juegos para sistemas LTI con acoplamientos dinámicos generalizados y restricciones convexas acopladas. El problema centralizado original es transformado en un juego dinámico de un número de problemas de optimización locales, los cuales se resuelven utilizando las variables de decisión relevantes para cada subsistema e intercambiando información con el objetivo de coordinar las decisiones. La relevancia del esquema de control distribuido propuesto radica en la capacidad de reducir la carga computacional y evitar los obstáculos de organización asociados con implementaciones centralizadas, mientras que retiene sus propiedades (estabilidad, optimalidad, factibilidad). En este contexto, el tipo de coordinación que se puede alcanzar es determinado por la conectividad y la capacidad de comunicación de la red, así como de la información disponible a partir de la función de costo y las restricciones del sistema. Se supondrá que la conectividad de la red de comunicación es suficiente para que los subsistemas obtengan información de todas las variables que aparecen en sus problemas locales. Se demostrará que en los casos que la función de costo y restricciones del sistema son conocidas por todos los controladores distribuidos, la solución del proceso iterativo converge a la solución del MPC centralizado. Esto significa que las propiedades (estabilidad, optimalidad, factibilidad) de la solución obtenida utilizando la implementación distribuida son las mismas que las de la solución utilizando la implementación centralizada. Finalmente, se estudian los efectos de fallas de comunicación en las propiedades del sistema (convergencia, estabilidad y desempeño). Se mostrarán los efectos de la partición del sistema y la comunicación en la convergencia y estabilidad, y se encontrará una cota superior al desempeño del sistema.

5.2. Control Predictivo Distribuido

5.2.1. Control predictivo

MPC se formula resolviendo un problema de control óptimo a lazo abierto on-line utilizando la estrategia de horizonte móvil. Usando el estado actual x_k , una secuencia de entradas de control \mathbf{u}_k es calculada con el objetivo de minimizar una función de costo o índice de desempeño $\mathcal{J}(x_k, \mathbf{u}_k)$ mientras que satisface las restricciones específicas. El primer elemento de la secuencia $u_{k|k}$ es aplicada al sistema y luego los horizontes de control y predicción se mueven una muestra hacia adelante al siguiente instante de muestreo. Se toman las nuevas mediciones para compensar las perturbaciones no medidas, que causan que la salida del sistema sea diferente a su predicción. Al instante k , el controlador resuelve el siguiente problema de optimización

$$\begin{aligned} \min_{\mathbf{u}_k \in \mathcal{U}} \mathcal{J}(k) \\ \text{st. } x_{k+1|k} = Ax_{k|k} + B\mathbf{u}_{k|k}. \end{aligned} \quad (5.1)$$

El Problema [5.1](#) puede ser reescrito en términos de la matriz de observabilidad Γ y la matriz de Haenkel \mathcal{H} del sistema [\[72\]](#)

$$\begin{aligned} \min_{\mathbf{u}_k \in \mathcal{U}} \mathcal{J}(k) \\ \text{st. } \mathbf{x}_{k+1} = \Gamma x_k + \mathcal{H}\mathbf{u}_k. \end{aligned} \quad (5.2)$$

donde la trayectoria de estados y entradas de control a tiempo k están dadas por

$$\begin{aligned} \mathbf{x}_k &= [x_{k|k}, \dots, x_{k+N|k}]^T, \\ \mathbf{u}_k &= [u_{k|k}, \dots, u_{k+N-1|k}]^T. \end{aligned}$$

Las variables $x_{k+i|k}$ y $u_{k+i|k}$ son los estados y entradas de control predichos a tiempo $k+i$ basados en la información a tiempo k y el modelo del sistema

$$x_{k+1} = Ax_k + Bu_k, \quad (5.3)$$

donde $x_k \in \mathfrak{R}^{n_x}$ y $u_k \in \mathcal{U} \subseteq \mathfrak{R}^{n_u}$. El conjunto de controles admisibles $\mathcal{U} = \{u \in \mathfrak{R}^{n_u} | Du \leq d, d > 0\}$ es un conjunto no vacío, compacto y convexo que contiene al origen en su interior.

Observación 5.1. El modelo centralizado definido en [\(5.3\)](#) es más general que el llamado modelo compuesto empleado en [\[129\]](#), el cual requiere que los estados de los subsistemas estén desacoplados y solo permite acoplamientos en las entradas. En este enfoque, el modelo centralizado puede tener estados y entradas de control acoplados.

En el problema de optimización (5.2), la función de costo $\mathcal{J}(x_k, \mathbf{u}_k)$ mide la diferencia entre el comportamiento predicho y la trayectoria deseada. Generalmente, se utiliza un índice cuadrático

$$\mathcal{J}(x_k, \mathbf{u}_k) = \sum_{i=0}^{N-1} \left\| x_{k+i|k} - x_{k+i}^{ref} \right\|_Q^2 + \left\| u_{k+i|k} \right\|_R^2 + \left\| x_{k+N|k} - x_{k+N}^{ref} \right\|_{P_{k|k}}^2 \quad (5.4)$$

Para garantizar la estabilidad a lazo cerrado, las matrices de peso satisfacen $Q > 0$, $R > 0$ y $P_{k|k}$ está dada por $A^T P_{k|k} A - P_{k|k} = -Q$ [72]. Para esta elección de matrices de peso, el índice (5.4) es equivalente a un índice de desempeño con horizonte infinito.

$$\mathcal{J}_\infty(x_k, \mathbf{u}_k) = \sum_{i=0}^{\infty} \left\| x_{k+i|k} - x_{k+i}^{ref} \right\|_Q^2 + \left\| u_{k+i|k} \right\|_R^2.$$

En varias formulaciones, una restricción o modos de control extra se incluyen en (5.2) para garantizar la estabilidad del sistema a lazo cerrado [72, 103].

5.2.2. MPC distribuido

Los sistemas de gran escala generalmente se componen de varios subsistemas que interactúan entre sí. Las interacciones pueden ser *a)* dinámicas, en el sentido que los estados y entrada de control de cada subsistema influyen los estados de los otros a los que está conectado, *b)* debido al hecho que los subsistemas comparten un objetivo y restricciones en común, o *c)* ambas. Los sistemas de este tipo admiten una descomposición en m subsistemas representados por

$$x_{k+1}^l = \sum_{p=1}^m A^{lp} x_k^p + B^{lj \in \mathcal{N}^p} u_k^{j \in \mathcal{N}^p} \quad l = 1, \dots, m \quad (5.5)$$

donde $x^l \in R^{n_{x^l}} \subseteq R^{n_x}$ y $u^l \in \mathcal{U}^l \subseteq R^{n_{u^l}} \subset R^{n_u}$ son los estados y entradas de control, respectivamente. El conjunto de superíndices de entradas de control del subsistema l se denota \mathcal{N}^l , y el conjunto \mathcal{I} denota todos los índices de entradas de control tal que $\mathbf{u}_k = u_k^{j \in \mathcal{I}}$.

Observación 5.2. *Este es una clase de modelos muy general para describir acoplamientos dinámicos entre subsistemas e incluye como caso especial la combinación de modelos descentralizados y de interacción, como en [129]. Los subsistemas pueden compartir las variables de entrada de manera que*

$$\sum_{l=1}^m n^{u_l} \geq n^u. \quad (5.6)$$

Supondremos que cada subsistema tiene restricciones locales convexas independientes y acopladas, las que involucran un pequeño número de los otros subsistemas. El conjunto de controles admisibles locales

$$\mathcal{U}^l = \left\{ u^l \in R^{n_{u^l}} \mid D^l u^l \leq d^l, d^l > 0 \right\} \quad (5.7)$$

también se supone no vacío, compacto, convexo y contiene el origen en su interior.

El esquema de control propuesto se basa en un conjunto de m agentes independientes que implementan optimizaciones de pequeña escala para el subsistema y se conectan a través de una red de comunicaciones de forma tal que pueden compartir los recursos comunes y coordinar sus acciones a fin de cumplir con los objetivos de control.

Suposición 5.1. *Los estados locales de cada subsistema x_k^l son accesibles.*

Suposición 5.2. *La comunicación entre los agentes de control es síncrona.*

Suposición 5.3. *Los agentes de control se comunican varias veces dentro de un intervalo de muestreo.*

Este conjunto de suposiciones no es restrictivo. De hecho, si los estados locales no se encuentran accesibles, pueden estimarse a partir de las salidas locales y_k^l utilizando un filtro de Kalman o MHE, por lo tanto la Suposición 5.1 es razonable. Asimismo, la Suposición 5.2 y Suposición 5.3 no son tan fuertes porque suponemos que el intervalo de muestreo es más largo que los tiempos de cálculo y comunicación.

Bajo estas suposiciones y descomposiciones, la función de costo (5.4) puede ser escrita como sigue

$$\mathcal{J}(x_k, \mathbf{u}_k, A) = \sum_{l=1}^m \alpha^l \mathcal{J}^l \left(x_k, \mathbf{u}_k^{j \in \mathcal{N}^l}, \mathbf{u}_k^{j \in \mathcal{I} - \mathcal{N}^l} \right), \quad (5.8)$$

donde $A = [\alpha^l]$, $\alpha^l \geq 0$, $\sum_{l=1}^m \alpha^l = 1$, \mathbf{u}_k^j es la trayectoria j -ésima de entradas del sistema. Esta descomposición de la función de costo y variables de entrada conduce a la descomposición (5.2) en m problemas de optimización acoplados

$$\begin{aligned} & \min_{\mathbf{u}_k^{j \in \mathcal{N}^l}} \mathcal{J}(x_k, \mathbf{u}_k, A) \\ & \text{st.} \\ & \mathbf{x}_{k+1} = \Gamma x_k + \mathcal{H} \mathbf{u}_k \\ & \mathbf{u}_k^{j \in \mathcal{N}^l} \in \mathcal{U}^{j \in \mathcal{N}^l} \\ & \mathbf{u}_k^{j \in \mathcal{I} - \mathcal{N}^l} \in \mathbf{U}^{j \in \mathcal{I} - \mathcal{N}^l} \end{aligned} \quad (5.9)$$

donde $\mathbf{U}^{j \in \mathcal{I} - \mathcal{N}^l}$ denota las entradas supuestas de los demás agentes. El objetivo de la descomposición es reducir la complejidad del problema de optimización (5.2) asegurando que los subproblemas (5.9) son más pequeños que el problema original (menos variables de decisión y restricciones), mientras que mantienen las propiedades del problema original. El precio a pagar por simplificar el problema de optimización (5.2) es la necesidad de coordinar entre los subproblemas (5.9) durante su solución. De esta forma, el problema de optimización (5.2) ha sido transformado en un *juego dinámico* de m agentes, donde cada uno busca sus decisiones óptimas mediante una secuencia de *juegos estratégicos* en respuesta a las decisiones de otros agentes.

Algoritmo 5.1

Dados $Q^l, R^l, 0 < q_{max} < \infty, \varepsilon^l > 0 \quad \forall l = 1, \dots, m$
Para cada agente $l, \quad l = 1, \dots, m$
Paso 1 Inicializar agente l
1.a Medir el estado local $x_k^l, q = 1,$
 $\rho^l = \phi \varepsilon^l \quad \phi \gg 1$
1.b $\mathbf{u}_k^0 = [u_{k|k-1}, \dots, u_{k+M|k-1} \ 0]$
Paso 2 **while** $\rho^l > \varepsilon^l$ and $q < q_{max}$
2.a Resolver el problema (5.9) para obtener ${}^q \tilde{\mathbf{u}}_k^{j \in \mathcal{N}^l}$
2.b **for** $p = 1, \dots, m$ and $p \neq l$
Comunicar ${}^q \tilde{\mathbf{u}}_k^{j \in \mathcal{N}^l}$ al agente p]
end
2.c Actualizar la solución iterada $q \quad \forall j \in \mathcal{N}^l$
 ${}^q \mathbf{u}_k^j = \sum_{p=1}^m \alpha^p {}^q \tilde{\mathbf{u}}_k^{i \in \mathcal{N}^p \cap j}$
 $+ \left(1 - \sum_{p=1}^m \alpha^p \text{card}(j \cap \mathcal{N}^l)\right) {}^{q-1} \mathbf{u}_k^j$
2.d $\rho^l = \left\| {}^q \mathbf{u}_k^{j \in \mathcal{N}^l} - {}^{q-1} \mathbf{u}_k^{j \in \mathcal{N}^l} \right\|_\infty$
 $q = q + 1$
end
Paso 3 Aplicar $u_{k|k}^l$
Paso 4 $k = k + 1$ e ir al Paso 1

Definición 5.1. Un juego dinámico $\langle m, \mathcal{U}, \mathcal{J}^l(x_k, {}^q \mathbf{u}_k, A), \mathcal{D}(q, k) \rangle$ modela la interacción de m agentes en q iteraciones y está compuesto de: i) $m \in \mathbb{N}$

agentes; ii) un conjunto no vacío \mathcal{U} que corresponde a las decisiones disponibles ${}^q\mathbf{u}_k^l$ de cada agente; iii) una función de utilidad $\mathcal{J}^l(x_k, {}^q\mathbf{u}_k, A) : x_k \times {}^q\mathbf{u}_k \rightarrow \mathbb{R}^+$ para cada agente; iv) un juego estratégico $\mathcal{G}(q, k)$ que modela la interacción entre agentes en la iteración q y tiempo k ; v) un proceso dinámico de ajuste de decisión $\mathcal{D}(q, k) : ({}^q\mathbf{u}_k, \mathcal{G}(q, k), q) \rightarrow {}^q\mathbf{u}_k$.

En cada etapa del juego dinámico, la decisión conjunta de los agentes determina el resultado del *juego estratégico* $\mathcal{G}(q, k)$ y cada agente tiene alguna preferencia ${}^q\mathbf{u}_k^{j \in \mathcal{N}^l}$ sobre el conjunto de posibles salidas \mathcal{U} . Basado en estos resultados y el proceso de ajuste $\mathcal{D}(q, k)$, el cual depende de la función de costo $\mathcal{J}^l(\cdot)$ y las restricciones, los agentes concilian sus decisiones. Más formal, un juego estratégico se define como sigue [88]

Definición 5.2. Un juego estratégico finito $\mathcal{G}(q, k) = \langle m, \mathcal{U}^l, \mathcal{J}^l(x_k, {}^q\mathbf{u}_k, A) \rangle$ modela las interacciones entre m agentes y está compuesto de: i) con conjunto finito no vacío $\mathcal{U}^l \subseteq \mathcal{U}$ que corresponde al conjunto de las decisiones disponibles para cada agente; ii) una función de utilidad $\mathcal{J}^l(x_k, {}^q\mathbf{u}_k, A) : x_k \times {}^q\mathbf{u}_k \rightarrow \mathbb{R}$ ${}^q\mathbf{u}_k \in \mathcal{U}$ para cada agente.

En general, estamos interesados en determinar las elecciones que los agentes realizarán cuando se encuentran con un juego en particular, a las cuales nos referiremos como la *solución del juego*. Adoptaremos el concepto más común de solución, conocido como *equilibrio de Nash* [86]: un conjunto de elecciones donde ningún agente mejora su utilidad por cambiar su elección de forma individual. De manera formal, tenemos:

Definición 5.3. Un grupo de decisiones de control \mathbf{u}_k se denomina Nash-óptimo si

$$\mathcal{J}^l \left(x_k, {}^q\mathbf{u}_k^{j \in \mathcal{N}^l}, {}^{q-1}\mathbf{u}_k^{j \in \mathcal{I} - \mathcal{N}^l} \right) \leq \mathcal{J}^l \left(x_k, {}^{q-1}\mathbf{u}_k^{j \in \mathcal{N}^l}, {}^{q-1}\mathbf{u}_k^{j \in \mathcal{I} - \mathcal{N}^l} \right)$$

donde $q > 0$ es el número de iteraciones transcurridas durante el proceso iterativo.

Si se ha alcanzado una solución Nash-óptima, cada subproblema no cambia su decisión ${}^q\mathbf{u}_k^{j \in \mathcal{N}^l}$ porque ha alcanzado un punto de equilibrio del proceso de decisiones; de otra manera, el índice de desempeño local \mathcal{J}^l se degradará. Cada subsistema optimiza su función objetivo utilizando su decisión de control propia ${}^q\mathbf{u}_k^{j \in \mathcal{N}^l}$, suponiendo que las soluciones de los demás subsistemas ${}^q\mathbf{u}_k^{j \in \mathcal{I} - \mathcal{N}^l}$ se conocen. Dado que la comunicación y el intercambio de información son tenidos en cuenta adecuadamente, cada subsistema resuelve su

problema de optimización siempre que las soluciones de los demás subsistemas son conocidas. Entonces, cada agente compara la nueva solución con la obtenida en la iteración previa y comprueba la condición de detención.

$$\left\| {}^q \mathbf{u}_k^{j \in \mathcal{N}^l} - {}^{q-1} \mathbf{u}_k^{j \in \mathcal{N}^l} \right\|_{\infty} \leq \varepsilon^l \quad l = 1, \dots, m. \quad (5.10)$$

Si el algoritmo es convergente, la condición (5.10) será cumplida por todos los agentes y el sistema completo va a alcanzar un punto de equilibrio. Los m subproblemas (5.9) pueden ser resueltos utilizando el Algoritmo 5.1.

En cada iteración k , q_{max} representa un límite por diseño en el número de iteraciones q y ε^l representa el criterio de detención del proceso iterativo. El usuario puede elegir detener el Algoritmo 1 antes de que se cumplan éstos límites.

5.3. Propiedades

5.3.1. Performance

Dado el esquema distribuido propuesto en la Sección previa, surgen tres preguntas fundamentales: *a*) el comportamiento de las iteraciones de los agentes durante el proceso de negociación, *b*) la ubicación y cantidad de puntos de equilibrio del problema distribuido, y *c*) la factibilidad de las soluciones. Uno de los factores clave en estas preguntas es el efecto de la función de costo y restricciones empleadas por los problemas distribuidos. Por lo tanto, en principio se explorará el efecto del índice de desempeño en la cantidad y ubicación de los puntos de equilibrio.

Primero, las condiciones de optimalidad del problema centralizado (5.2) serán desarrolladas a fin de tener una medida de comparación con el desempeño de los esquemas de control distribuidos. Para hacer la comparación más sencilla, el índice de desempeño (5.4) se descompone en m componentes relacionados a los subsistemas, como en los problemas distribuidos (5.9), de la siguiente manera

$$\mathcal{J}(x_k, \mathbf{u}_k, \Theta) = \sum_{l=1}^m \theta^l \mathcal{J}^l(x_k, \mathbf{u}_k), \theta^l \geq 0, \sum_{l=1}^m \theta^l = 1. \quad (5.11)$$

Esta forma de escribir el índice de desempeño corresponde a una caracterización multiobjetivo del problema (5.2). Al aplicar las condiciones de opti-

malidad de primer orden, obtenemos

$$\sum_{l=1}^m \theta^l \frac{\partial \mathcal{J}^l(x_k, \mathbf{u}_k)}{\partial \mathbf{u}_k^{j \in \mathcal{N}^p}} + \lambda^T D^{j \in \mathcal{N}^p} = 0 \quad p = 1, \dots, m \quad (5.12a)$$

$$\lambda^T \left(D^{j \in \mathcal{N}^p} \mathbf{u}_k^{j \in \mathcal{N}^p} - b \right) = 0 \quad (5.12b)$$

donde D^j es el j -ésimo vector columna de D . La solución de este conjunto de ecuaciones \mathbf{u}_k^* es la solución óptima del problema de optimización (5.2) y pertenece a un *conjunto de Pareto*, el cual se define como [43].

Definición 5.4. Una solución $\mathbf{u}_k^* \in \mathcal{U}$ se dice Pareto-óptima del problema de optimización (5.2) si no existe otra solución factible $\forall \mathbf{u}_k \in \mathcal{U}$ tal que $\mathcal{J}^l(x_k, \mathbf{u}_k) \leq \mathcal{J}^l(x_k, \mathbf{u}_k^*) \quad \forall l = 1, \dots, m$.

En control distribuido, los agentes coordinan sus decisiones mediante un proceso de negociación. Al aplicar las condiciones de optimalidad de primer orden al costo descentralizado (5.8) obtenemos

$$\sum_{l=1}^m \alpha^l \frac{\partial \mathcal{J}^l(x_k, \mathbf{u}_k)}{\partial \mathbf{u}_k^{j \in \mathcal{N}^p}} + \lambda^T D^{j \in \mathcal{N}^p} = 0 \quad p = 1, \dots, m \quad (5.13a)$$

$$\lambda^T \left(D^{j \in \mathcal{N}^p} \mathbf{u}_k^{j \in \mathcal{N}^p} - b \right) = 0. \quad (5.13b)$$

Por simple inspección de (5.12) y (5.13) podemos ver que estas ecuaciones tienen la misma estructura, solo difieren en sus pesos. Por lo tanto, la ubicación de los puntos de equilibrio de los esquemas distribuidos va a depender en la selección de α^l con $l = 1, \dots, m$. Hay dos opciones:

- Si $\alpha^l = 1 \quad \forall l, \alpha_{p \neq l} = 0$ la condición de optimalidad (5.13) se vuelve

$$\frac{\partial \mathcal{J}^l(x_k, \mathbf{u}_k)}{\partial \mathbf{u}_k^{j \in \mathcal{N}^l}} + \lambda^T D^{j \in \mathcal{N}^l} = 0 \quad l = 1, \dots, m \quad (5.14a)$$

$$\lambda^T \left(D^{j \in \mathcal{N}^l} \mathbf{u}_k^{j \in \mathcal{N}^l} - b \right) = 0. \quad (5.14b)$$

Esta condición solo evalúa el efecto de $\mathbf{u}^{j \in \mathcal{N}^l}$, dado $\mathbf{u}^{j \in \mathcal{I} - \mathcal{N}^l}$, en el subsistema l sin tener en cuenta sus efectos en los agentes restantes (**comportamiento egoísta**). Esta configuración del problema distribuido conduce a un **juego incompleto de información perfecta** que puede alcanzar soluciones Nash-óptimas para una estrategia pura (**equilibrio de Cournot**) [88]. Al comparar (5.12) y (5.14) podemos

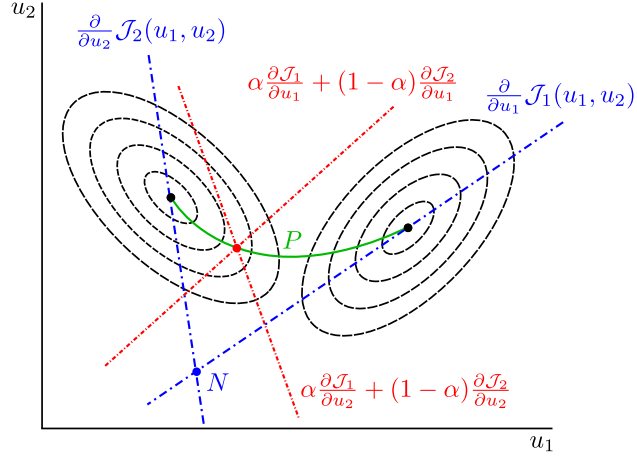


Figura 5.1: Visualización de las condiciones de optimalidad generadas por las Ecuaciones (5.13).

concluir que la solución de estas ecuaciones está afuera del conjunto de Pareto [32, 87]. La razón de la ineficiencia del equilibrio de Nash radica en el hecho que la información de los efectos de las variables de decisión en los agentes restantes es omitida ($\alpha_{p \neq l} = 0$ juego de información incompleta). Por consiguiente, cada agente minimiza su índice de desempeño, acomodando los efectos de las decisiones de los otros agentes, sin tener en cuenta sus efectos en el resto del sistema. Además de la falta de optimalidad, la cantidad de puntos de equilibrio generados por la condición de optimalidad (5.14) puede crecer con el número de agentes.

- Si $\alpha^l > 0 \forall l$, la condición de optimalidad (5.13) se vuelve

$$\sum_{l=1}^m \alpha^l \frac{\partial \mathcal{J}^l(x_k, \mathbf{u}_k)}{\partial \mathbf{u}_k^{j \in \mathcal{N}^p}} + \lambda^T D^{j \in \mathcal{N}^p} = 0 \quad p = 1, \dots, m \quad (5.15a)$$

$$\lambda^T \left(D^{j \in \mathcal{N}^p} \mathbf{u}_k^{j \in \mathcal{N}^p} - b \right) = 0. \quad (5.15b)$$

Esta condición evalúa el efecto de $\mathbf{u}^{j \in \mathcal{N}^l}$ dado $\mathbf{u}^{j \in \mathcal{I} - \mathcal{N}^l}$ en el sistema completo, teniendo en cuenta el efecto de las interacciones entre los subsistemas (**comportamiento cooperativo**), lo que conduce a un **juego completo de información perfecta**. Al comparar (5.12) y (5.15) podemos ver que estas dos ecuaciones tienen estructura similar, por lo que podemos concluir que sus soluciones están dentro del conjunto de Pareto. La posición de las soluciones del problema de MPC

distribuido dependerá de los valores de α^l . En el caso particular de $\alpha^l = \theta^l \quad l = 1, \dots, m$ la solución del problema centralizado y del distribuido son las mismas.

El valor de los pesos $\alpha^l \quad l = 1, \dots, m$ depende de la estructura de la información; esto es la información de la función de costo y restricciones disponibles en cada agente. Si la función de costo y restricciones de cada agente son conocidas por todos los demás, α^l puede elegirse como el segundo esquema distribuido ($\alpha^l > 0 \quad \forall l = 1, \dots, m$). En este caso, el problema de optimización es distribuido entre m agentes independientes que coordinan sus soluciones con el objetivo de resolver el problema de optimización de forma distribuida. Por este motivo, este esquema de control es denominado **MPC distribuido**. En el otro caso, cuando las funciones de costo local y restricciones son solamente conocidas por los agentes, los pesos α^l deberían ser elegidos como el primer esquema ($\alpha^l = 1, \alpha_{p \neq l} = 0 \quad \forall l, p = 1, \dots, m$). En este caso, el problema de optimización centralizado es descentralizado en m agentes independientes que sólo coordinan los efectos de sus decisiones con el objetivo de minimizar el efecto de las interacciones. Por este motivo, llamamos a este esquema de control **MPC descentralizado coordinado**.

Observación 5.3. *El hecho que los agentes alcancen la optimalidad según el criterio de Nash no implica que se haya alcanzado el óptimo global de la solución. Esta relación dependerá en la estructura de la función de costo de los agentes y las restricciones, las cuales dependen en el valor de los pesos α^l y el número de iteraciones permitidas.*

La estructura de $\mathbf{u}^{j \in \mathcal{N}^l}$ determina la estructura de las restricciones que pueden manejar los esquemas distribuidos. Si los subproblemas comparten las variables de entrada involucradas en las restricciones acopladas ($\mathcal{N}^l \cap \mathcal{N}^{p \neq l} \neq \emptyset$), el esquema MPC distribuido puede resolver problemas de optimización con restricciones acopladas. Por otro lado, cuando los subproblemas no incluyen las variables de entrada de las restricciones acopladas ($\mathcal{N}^l \cap \mathcal{N}^{p \neq l} = \emptyset$), los esquemas MPC distribuidos solo resuelven problemas de optimización con restricciones independientes [34, 52, 129]. Estos hechos se vuelven aparentes a partir de las condiciones de optimalidad (5.14) y (5.15).

5.3.2. Convergencia

Los subproblemas (5.9) pueden competir o cooperar en la solución del problema global. El comportamiento de cada agente dependerá de la existencia, o no, de objetivos conflictivos que pueden surgir de las características de las interacciones: los objetivos de control y las restricciones. La manera

en que el sistema es descompuesto es uno de los factores que define el comportamiento del durante las iteraciones, ya que define cómo las interacciones serán abordadas por los esquemas distribuidos.

El sistema global puede ser particionado de acuerdo a la estructura del sistema físico o siguiendo las bases del análisis del modelo matemático, o una combinación de ambas. Se han propuesto procedimientos heurísticos para el particionado del sistema basado en análisis de entrada–salida (ver [42, 47, 126]), basado en espacio de estados (ver [105, 131]) o en el desempeño métrico para el particionado óptimo de sistemas de control distribuidos y jerárquicos (ver [50, 81]). En todos estos enfoques, el objetivo es simplificar el diseño de control reduciendo los acoples dinámicos, de forma tal que los requerimientos computacionales se distribuyan de forma uniforme para evitar una carga de comunicación excesiva. Es importante notar que el particionado de un modelo en espacio de estados puede conducir a estados solapados debido a las dinámicas acopladas en el sistema continuo y debido al muestreo en tiempo discreto, que puede cambiar la estructura *sparse* en el modelo.

Suposición 5.4. *El modelo utilizado por los algoritmos MPC distribuido son particionados siguiendo los procedimientos descritos en [81].*

Para analizar el efecto de la descomposición del sistema en el esquema distribuido con restricciones, primero analizaremos sus efectos en el problema sin restricciones. Al resolver la condición de optimalidad (5.13) para un sistema sin restricciones, nos conduce a

$${}^q \mathbf{u}_k = \mathcal{K}^0 {}^{q-1} \mathbf{u}_k + \mathcal{K}^1 x_k \quad \forall q > 0, \quad (5.16)$$

la cual modela el comportamiento del problema distribuido durante el proceso iterativo. Su estabilidad induce la convergencia del proceso iterativo y está dada por

$$\left| \lambda(\mathcal{K}^0) \right| < 1. \quad (5.17)$$

La ganancia \mathcal{K}^1 es el controlador descentralizado que calcula la contribución de x_k en \mathbf{u}_k y tiene elementos distintos a cero solo en su diagonal $\mathcal{K}^1 = [\mathcal{K}^{ll}] \quad l = 1, \dots, m$. Por otro lado, \mathcal{K}^0 modela la interacción entre subsistemas durante el proceso iterativo, determinando su estabilidad y tiene elementos distintos de cero fuera de la diagonal.

$$\mathcal{K}^0 = \begin{bmatrix} 0 & \mathcal{K}^{12} & \dots & \mathcal{K}^{1m} \\ \mathcal{K}^{21} & 0 & & \mathcal{K}^{2m} \\ \vdots & & \ddots & \vdots \\ \mathcal{K}^{m1} & \dots & \mathcal{K}^{mm-1} & 0 \end{bmatrix}. \quad (5.18)$$

La estructura de los componentes de \mathcal{K}^0 y \mathcal{K}^1 depende del valor de los pesos α^l :

- Si se adopta el *MPC coordinado descentralizado* ($\alpha^l = 1, \alpha_{p \neq l} = 0$) los elementos d \mathcal{K}^0 están dados por

$$\mathcal{K}^{lp} = -K^l \mathcal{H}^{lp} \quad l, p = 1, \dots, m \quad (5.19)$$

donde $K^l = \left(\mathcal{H}^{lT} Q^l \mathcal{H}^l + R^l \right)^{-1} \mathcal{H}^{lT} Q^l$. Por lo tanto, la manera en que el problema global (5.2) fue particionado y la forma en que los parámetros del controlador fueron sintonizados definen la convergencia del *MPC coordinado descentralizado*. Si se cumple la Suposición 5.2, la convergencia del algoritmo puede garantizarse para aquellos sistemas que tengan interacciones débiles.

- Por otro lado, cuando se emplea *MPC distribuido* ($\alpha^l > 0$) la ganancia \mathcal{K}^0 está dada por

$$\mathcal{K}^{lp} = -K^{lp} \mathcal{H}^{lp} \quad l, p = 1, \dots, m \quad (5.20)$$

donde $K^{lp} = \left(\mathcal{H}^{lpT} Q^{lp} \mathcal{H}^{lp} + R^{lp} \right)^{-1} \mathcal{H}^{lpT} Q^{lp}$ son las ganancias del controlador. Dada que *MPC distribuido* está diseñado para garantizar la estabilidad del sistema completo, su convergencia está garantizada independientemente de la manera de que el sistema haya sido particionado.

Ahora, se considerarán sistemas con restricciones. En este caso, si se cumple la Suposición 5.2, la convergencia de los sistemas con restricciones puede ser analizada utilizando argumentos de Lyapunov. La idea principal es mostrar la contractividad de la secuencia de funciones de costo globales $\mathcal{J}(x_k, \mathbf{u}_k, A)$ generadas por el Algoritmo 5.2.2 a lo largo del proceso iterativo.

Lema 5.1. *Supongamos que el sistema ha sido particionado siguiendo un procedimiento de diseño descentralizado y los problemas MPC distribuidos (5.9) $\forall l = 1, \dots, m$ son factibles, entonces la secuencia de funciones de costo $\mathcal{J}(x_k, \mathbf{u}_k, A)$ generadas por el Algoritmo 5.2.2 durante el proceso iterativo es no creciente $\forall q > 0$ en cualquier tiempo k .*

Demostración. Ver apéndice C.1. □

5.3.3. Factibilidad

A pesar que en la literatura generalmente se supone que existe y está disponible una solución inicial centralizada factible, en esta Sección se proveerá una forma simple e implementable de construirla en forma distribuida, suponiendo que el estado inicial global está disponible de antemano.

Una solución inicial factible ${}^0\mathbf{u}_k^{j \in \mathcal{N}^l}$ a tiempo $k = 0$ puede ser calculada localmente utilizando una aproximación interior del conjunto global factible \mathcal{U} basada en todas las restricciones que aparecen en (5.2) y el estado inicial global x_0 , el cual se supone disponible. Aproximemos la región factible con un poliedro circunscripto Ω de \mathcal{U} , la cual toma la forma del producto Cartesiano

$$\Omega = \Omega^1 \times \dots \times \Omega^m \subset \mathcal{U}. \quad (5.21)$$

Esta aproximación esencialmente descompone y desacopla las restricciones de los subsistemas al realizar *constraint tightening*. Cada subsistema l deberá incluir Ω^l en la configuración local del problema. Dado que el producto Cartesiano de estos conjuntos de restricciones locales está incluido en el conjunto global factible \mathcal{U} , cualquier combinación de soluciones globales dentro Ω^l será globalmente factible, también. Los conjuntos de restricciones locales que surgen de esta aproximación mediante un poliedro circunscripto serán, en general, bastante conservadores, pero al mismo tiempo, permitirán la construcción de una solución factible localmente que permita inicializar el Algoritmo 5.2.2.

El cálculo del poliedro circunscripto puede realizarse a priori y las restricciones locales Ω^l pueden ser distribuidas a cada subsistema. El cálculo de la caja de volumen máximo puede seguir el procedimiento descrito en [10]. A partir de esa instancia, es sencillo obtener las restricciones locales Ω^l . Para cada $k > 0$, se construye una solución factible al ejecutar el **Paso 1** del Algoritmo 5.2.2

$${}^0\mathbf{u}_k^{j \in \mathcal{N}^l} = \left[u_{k|k-1}^{j \in \mathcal{N}^l}, \dots, u_{k+M|k-1}^{j \in \mathcal{N}^l}, 0 \right]$$

La factibilidad durante las iteraciones se mantiene porque en el **Paso 2.a** se obtienen m soluciones factibles ${}^q\tilde{\mathbf{u}}_k^{j \in \mathcal{N}^l}$. Entonces, en el **Paso 2.c**, el nuevo perfil de control ${}^q\mathbf{u}_k^{j \in \mathcal{N}^l}$ se construye como una combinación convexa de estas soluciones. Dado que el problema (5.9) es un problema QP convexo con restricciones, cualquier combinación convexa de ${}^q\mathbf{u}_k^{j \in \mathcal{N}^l}$ satisface el conjunto convexo de restricciones. Por lo tanto, ${}^q\mathbf{u}_k$ es una solución factible del problema de optimización (5.9) para todo l .

5.3.4. Estabilidad

La demostración de la estabilidad nominal del sistema a lazo cerrado resultante utiliza argumentos estándar en su mayoría [79]. La demostración en esta sección está relacionada a la demostración de estabilidad del método MPC en [129]; con la adición de la Suposición 5.2. La idea principal es mostrar la contractividad de la secuencia de funciones de costo globales generadas por el Algoritmo 5.2.2 durante de la operación del sistema y la estabilidad del origen.

Teorema 5.1. *Supongamos que el sistema ha sido particionado siguiendo un procedimiento de diseño descentralizado y que el problema de optimización (5.9) es factible y ha sido resuelto utilizando el Algoritmo 1, entonces el origen es un punto de equilibrio exponencialmente estable.*

Demostración. Ver apéndice C.2. □

5.4. Comportamiento bajo fallas de comunicación

En el esquema propuesto, los agentes coordinan sus acciones mediante el intercambio de información utilizando la red de comunicaciones. Dado que los agentes utilizan la red de comunicaciones de forma extensiva, surgen algunas preguntas sobre el comportamiento del sistema en caso de que las comunicaciones fallen: ¿Cuáles son las condiciones para la convergencia del proceso iterativo? ¿Cómo es afectada la estabilidad a lazo cerrado? ¿Como cambia el desempeño del sistema?

En una primera etapa, las fallas en el sistema de comunicación son modeladas mediante tres matrices: *i*) la *matriz de conexión* \mathcal{C} , la cual representa la estructura de comunicaciones, *ii*) la *matriz de fallas de transmisión* \mathcal{T} , la cual modela fallas en la transmisión, y *iii*) la *matriz de fallas de recepción* \mathcal{R} , la cual modela las fallas de recepción en el sistema. La matriz \mathcal{C} se define como

$$\mathcal{C} = [c^{lp}], \quad c^{lp} = \begin{cases} 0 & l = p, \\ 1 \text{ or } 0 & l \neq p, \end{cases} \quad (5.22)$$

donde $c^{lp} = 1$ indica la conexión entre los agentes l y p , mientras que $c^{lp} = 0$ establece que no hay conexión entre estos agentes. Entonces, los fallos en la comunicación pueden ser modelados mediante la combinación de la matriz de conexión con las otras matrices que modelan las fallas en la recepción (\mathcal{R})

y transmisión (\mathcal{T}), denominada \mathcal{RCT} , la cual está dada por

$$\mathcal{R} = [r^{lp}], \quad r^{lp} = \begin{cases} 1 & l = p, \\ 0 & l \neq p, \end{cases} \quad (5.23a)$$

$$\mathcal{T} = [t^{lp}], \quad t^{lp} = \begin{cases} 1 & l = p, \\ 0 & l \neq p. \end{cases} \quad (5.23b)$$

Un elemento $t^{ll} = 1$ ($r^{ll} = 1$) corresponde a una transmisión (recepción) perfecta del agente l , mientras que $t^{ll} = 0$ ($r^{ll} = 0$) corresponde a una falla de transmisión (recepción) del agente l . Una falla entre agentes l y p es representada con la transición $1 \rightarrow 0$ de los elementos correspondientes de \mathcal{R} y \mathcal{T} .

Siguiendo el mismo procedimiento que en la Sección [5.3.2](#) la solución para el problema distribuido en cada iteración es

$${}^q \mathbf{u}_k = (\mathcal{K}^0 \mathcal{RCT})^{q-1} \mathbf{u}_k + (\mathcal{K}^1 \mathcal{RCT}) x_k \quad q \geq 1$$

Su comportamiento está relacionado con su estabilidad, la cual está dada por

$$\left\| \lambda(\mathcal{K}^0 \mathcal{RCT}) \right\|_1 < 1 \quad (5.24)$$

Si hay fallas de comunicación, cada agente no podrá intercambiar la información de manera adecuada, lo que modifica el proceso iterativo y lo conduce a obtener otra solución. En este caso, el agente con fallas de comunicación se convertirá en un controlador descentralizado que recibirá información sobre la decisión de los demás agentes a través del sistema. Esto deteriorará los márgenes de estabilidad y el desempeño debido a la presencia de interacciones que no fueron tenidas en cuenta durante el proceso iterativo, que actuarán como perturbaciones no medibles. En el caso extremo $\mathcal{K}^0 \mathcal{RCT} = 0$, la estructura de control se corresponderá con una *arquitectura de control completamente descentralizada*, y la estabilidad dependerá de la forma en que el sistema fue particionado. Si los controladores fueron diseñados siguiendo un procedimiento de diseño descentralizado [\[131\]](#), se puede garantizar la estabilidad del sistema a lazo cerrado.

Una vez que la convergencia de las iteraciones puede ser garantizada, el siguiente problema a abordar es el efecto de las fallas de comunicación en la estabilidad a lazo cerrado. Con el objetivo de establecer la estabilidad en el comportamiento a lazo cerrado, la acción de control

$$\tilde{\mathbf{u}}_k = (I - \mathcal{K}^0 \mathcal{RCT})^{-1} \mathcal{K}^1 \mathcal{RCT} \Gamma x_k$$

es reemplazada en el modelo a lazo abierto del sistema, conduciendo al sistema a lazo cerrado

$$x_{k+1} = \left(A - B\mathcal{I} (I - \mathcal{K}^0 \mathcal{R} \mathcal{C} \mathcal{T})^{-1} \mathcal{K}^1 \mathcal{R} \mathcal{C} \mathcal{T} \Gamma \right) x_k.$$

Luego, la estabilidad del sistema a lazo cerrado sujeto a fallas de comunicación está determinada por

$$\left| \lambda \left(A - B\mathcal{I} (I - \mathcal{K}^0 \mathcal{R} \mathcal{C} \mathcal{T})^{-1} \mathcal{K}^1 \mathcal{R} \mathcal{C} \mathcal{T} \Gamma \right) \right| < 1. \quad (5.25)$$

Bajo fallas de comunicación, cada agente no puede intercambiar información de forma adecuada y por lo tanto la estabilidad del sistema a lazo cerrado dependerá en las características dinámicas de las interacciones entre subsistemas. En el caso extremo $\mathcal{R} \mathcal{C} \mathcal{T} = 0$, la condición de estabilidad se satisface siempre, lo que corresponde a una arquitectura completamente descentralizada. Las interacciones actúan como perturbaciones no medibles para los controladores, reduciendo los márgenes de estabilidad y degradando el desempeño del sistema.

Teorema 5.2. *Supongamos que el sistema ha sido particionado de una forma en la que se cumple la condición de convergencia (5.17), su desempeño al instante k bajo fallas de comunicaciones locales es*

$$\tilde{\mathcal{J}}(k) \leq \left(1 + \frac{\|\mathcal{W}(k)\|}{\lambda_{\min}(\mathcal{F})} \right) \mathcal{J}^*, \quad (5.26)$$

donde la degradación del desempeño está acotada por

$$\frac{\tilde{\mathcal{J}}(k) - \mathcal{J}^*}{\mathcal{J}^*} \leq \frac{\|\mathcal{W}_{\max}\|}{\lambda_{\min}(\mathcal{F})} \quad (5.27)$$

donde $\lambda_{\min}(\mathcal{F})$ denota el autovalor mínimo de

$$\begin{aligned} \mathcal{F} &= \left[(\mathcal{K}^1)^{-1} (I - \mathcal{K}^0) - \mathcal{H} \right]^T Q \left[(\mathcal{K}^1)^{-1} (I - \mathcal{K}^0) - \mathcal{H} \right] + R, \\ \mathcal{W}_{\max} &= \mathcal{S}_{\max}^T \left(\mathcal{H}^T Q \mathcal{H} + R \right) \mathcal{S}_{\max}, \\ \mathcal{S}_{\max} &= 2I - \left[I + (I - \mathcal{K}^0)^{-1} (I + \mathcal{K}^0) \right]^{-1}. \end{aligned}$$

Demostración. Ver apéndice C.3. □

5.5. Resultados

En esta sección se muestra la aplicabilidad de los resultados teóricos presentados en este capítulo mediante simulación utilizando de múltiples vehículos terrestres a fines de ilustrar el desempeño del controlador descentralizado. El modelo utilizado por cada uno de los vehículos es

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{v}_x \\ \dot{v}_y \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \end{bmatrix}, \quad (5.28)$$

el cual fue discretizado utilizando un período de muestreo de 0.5 segundos. A la formulación de los controladores MPC centralizado y descentralizado se les incorporó las restricciones necesarias para que los vehículos mantengan distancia,

$$\rho^2 - d(R0, R1)^2 + s_0 \leq 0 \quad (5.29)$$

$$\rho^2 - d(R0, R2)^2 + s_1 \leq 0 \quad (5.30)$$

$$\rho^2 - d(R1, R2)^2 + s_2 \leq 0 \quad (5.31)$$

donde ρ es el radio de separación entre vehículos, el cual fue configurado en 0.4 m. El operador distancia se define como

$$d(R_i, R_j) = \sqrt{(x_{R_i} - x_{R_j})^2 + (y_{R_i} - y_{R_j})^2} \quad (5.32)$$

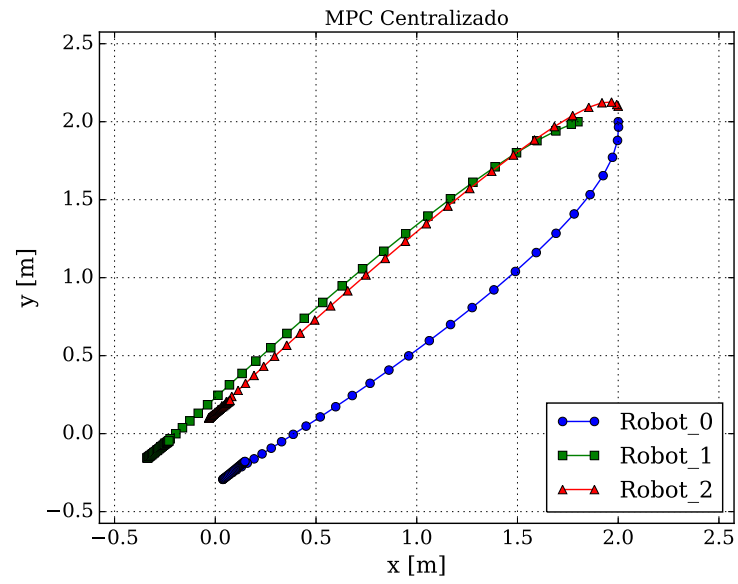
y las variables s_0 , s_1 y s_2 son variables de *slack* del problema de optimización, cuya solución fue obtenida mediante una sola iteración para hacer mas eficiente la ejecución del algoritmo. Finalmente, la configuración de los algoritmos de control predictivos es $Q = \text{diag}([5, 5, 0, 0])$, $R = \text{diag}([50, 50, 10000, 10000])$, $P_{k|k} = \text{diag}([50, 50, 0, 0])$ y $N = 10$.

La Figuras 5.2a y 5.2b muestran la posición de los vehículos en el plano x - y al utilizar MPC centralizado y descentralizado, respectivamente. Aquí se puede observar que las trayectorias obtenidas son diferentes, lo cual puede atribuirse a la cantidad de iteraciones seleccionadas. En ambos casos los vehículos se dirigen al origen de coordenadas y llegan a un equilibrio entre mantener la distancia mínima entre ellos y acercarse a la posición objetivo. En la Figura 5.3 se puede observar en detalle la posición x e y y las acciones de control de cada uno de los vehículos con respecto al tiempo. Aquí podemos observar, como era de esperarse, la similitud entre las acciones de control de los dos esquemas utilizados. Finalmente, la Figura 5.4 muestra la distancia

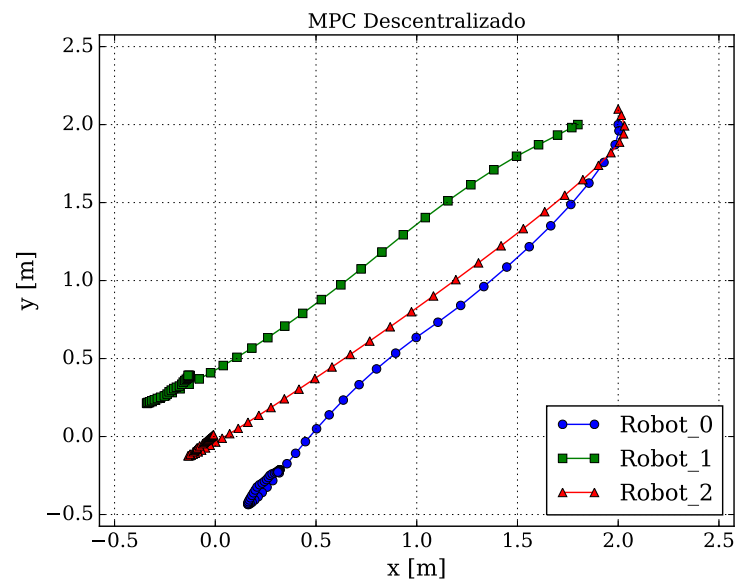
entre los vehículos. Aquí se puede observar que inicialmente la distancia ρ no se respeta, por lo que los controladores tienden a alejar los vehículos para luego tratar de mantener una distancia de al menos ρ entre ellos.

5.6. Conclusiones

En este capítulo se presentó un algoritmo de control predictivo distribuido basado en juegos dinámicos. El MPC es implementado de forma distribuida con agentes que resuelven el problema descentralizado equivalente en un entorno de red. Estos agentes pueden cooperar y comunicarse entre ellos para alcanzar el objetivo del sistema completo. Los efectos de los acoplamientos entre los agentes son tenidos en cuenta en este esquema, por lo que se logra un mejor desempeño que otros métodos tradicionales de control descentralizado. La principal ventaja de este esquema es que el problema de optimización online puede convertirse en un problema de optimización de varios sistemas de menor escala, por lo que se reduce significativamente la carga computacional mientras se cumplen los requerimientos de desempeño. Además, los parámetros de diseño para cada agente –tales como el horizonte de predicción y de control, las matrices de peso, el intervalo de muestreo, entre otros– pueden ser elegidos y sintonizados de forma separada, lo que provee más flexibilidad para el análisis y su aplicación en subsistemas de distintas características.



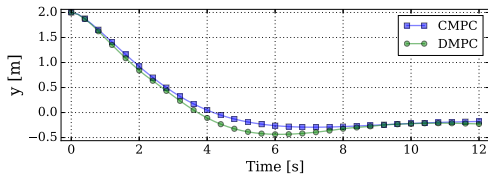
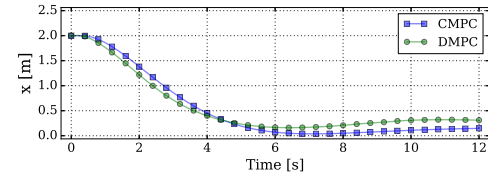
(a)



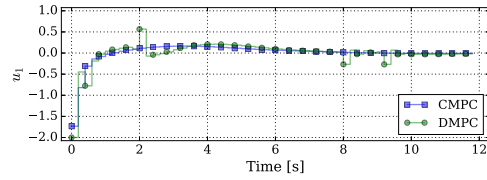
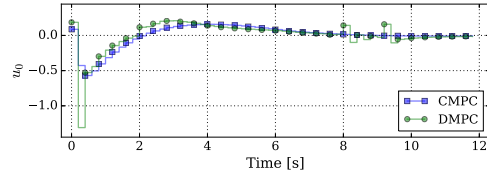
(b)

Figura 5.2: Posiciones de los vehículos utilizando a) MPC centralizado y b) MPC descentralizado.

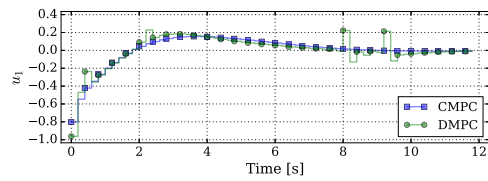
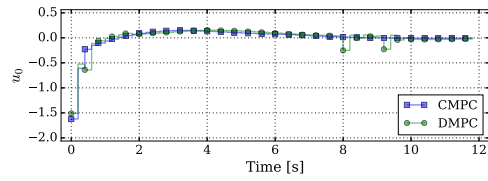
Conclusiones



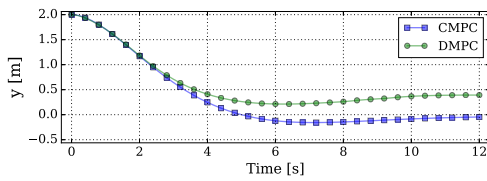
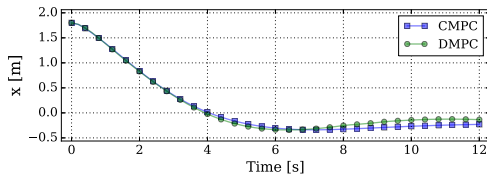
(a) Posiciones x e y del Robot 0.



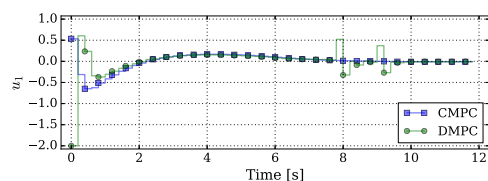
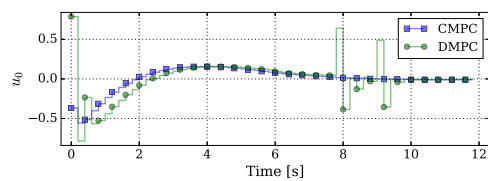
(b) Acciones de control del Robot 0.



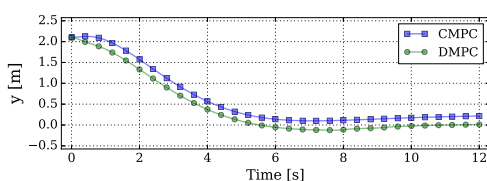
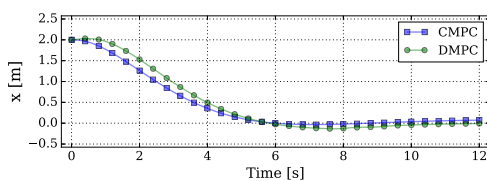
(c) Acciones de control del Robot 1.



(d) Posiciones x e y del Robot 1.

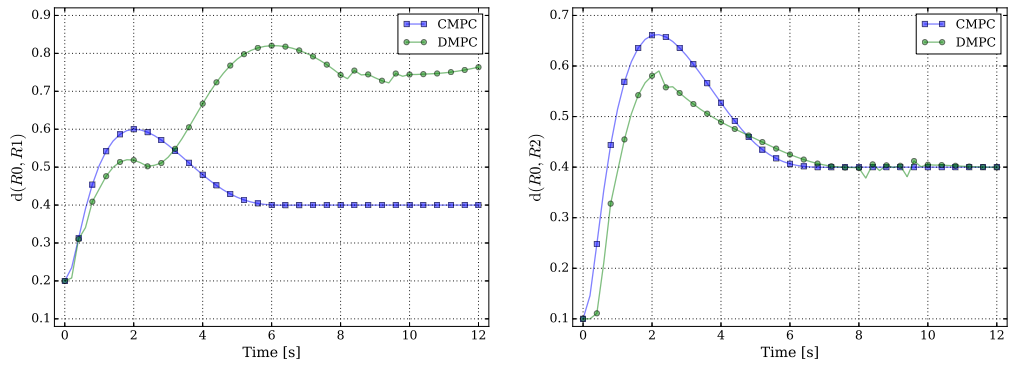


(e) Acciones de control del Robot 2.



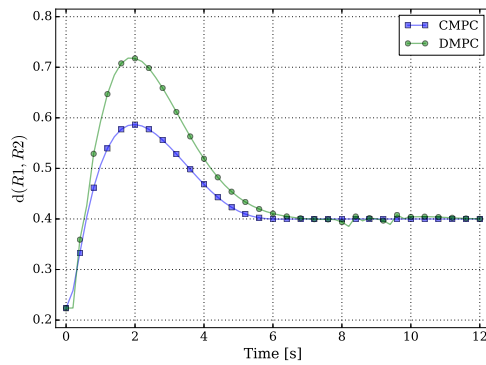
(f) Posiciones x e y del Robot 2.

Figura 5.3: Posiciones y acciones de control de los vehículos utilizando MPC centralizado y MPC descentralizado.



(a) Distancia entre los robots 0 y 1.

(b) Distancia entre los robots 0 y 2.



(c) Distancia entre los robots 1 y 2.

Figura 5.4: Distancia entre los robots obtenidas por MPC centralizado y descentralizado.

Capítulo 6

Conclusiones y trabajos futuros

6.1. Conclusiones

El objetivo de esta tesis fue controlar un sistema multiagente conformado por varios vehículos. Para ello, es necesario ser capaz de estimar los estados y controlar a cada agente de forma individual antes de pretender controlar un grupo de vehículos. En este aspecto, cabe destacar que en la gran mayoría de la bibliografía consultada, aquellos que realizan trabajos de estimación no hacen control y aquellos que realizan trabajos de control, no hacen estimación. Es por ello, que se hace hincapié en el desarrollo de algoritmos de estimación y control que permitan estimar los estados y controlar a un vehículo.

De esta forma, se propuso un marco unificado basada en modelos y métodos de horizonte móvil para resolver los problemas de **navegación** y **guiado** para un vehículo. La utilización de problemas de optimización permite incorporar las restricciones propias del sistema a los procesos de estimación y control. La técnica de linealización a lo largo de una trayectoria predefinida en conjunto con una buena aproximación del costo de arribo, permite reducir el horizonte de estimación y de control, reduciendo así la carga computacional de los algoritmos propuestos.

Debido a que los algoritmos utilizan una descripción del vehículo en forma de un modelo matemático, se han estudiado los avances en este campo y se presentó una forma sistemática para obtener modelos cinemáticos y dinámicos para WMR de las más diversas características y geometrías. Esta técnica se utilizó para generalizar el modelado de un robot de tracción diferencial en ambientes 2D y 3D, dando así el puntapié inicial para considerar WMR que sean capaces de moverse en terreno arbitrario en ambientes exteriores.

En cuanto a los algoritmos de MHE, cabe destacar que la aproximación adaptativa del costo de arribo permite obtener buenas aproximaciones de

este término y la capacidad de adaptación de los factores de olvido variable permiten que el estimador incorpore información relevante a partir de los datos. Gracias a esto, se puede reducir el horizonte de estimación y estar más cerca de la operación en tiempo real.

Los algoritmos NLMPC propuestos son estables y robustos, donde la optimalidad, estabilidad y factibilidad están garantizadas. Estos algoritmos han demostrado ser capaces de resolver tanto el problema de guiado como el de planeamiento de trayectoria utilizando horizontes de control reducidos. Mediante linealizaciones sucesivas sobre una trayectoria, se reduce la carga computacional producto de resolver un QP equivalente en vez de un NLP.

Para el caso de múltiples vehículos, se estudió la equivalencia entre un problema de control centralizado y uno distribuido basado en juegos dinámicos. El MPC distribuido permite que los vehículos resuelvan un problema equivalente en un entorno de red, que les permite cooperar y comunicarse entre ellos para alcanzar el objetivo del sistema completo. De esta forma, si somos capaces de controlar un vehículo, podemos ser capaces de controlar varios y realizar formaciones en misiones donde es necesario que los vehículos sean capaces de lograr de forma independiente el objetivo grupal, aún ante la presencia de posibles fallas de comunicación.

La utilización de CasADi posibilita la implementación de algoritmos rápidos de optimización utilizando Python, lo que permite la operación en tiempo real utilizando un lenguaje de programación interpretado, multiplataforma y de buena performance.

6.2. Trabajos futuros

El campo de la estimación basada en horizonte móvil se encuentra menos estudiado que el de control y ofrece más posibilidades de desarrollo a futuro. Una de estas posibilidades es el desarrollo de algoritmos de estimación robusta basados en conjuntos para estimar la posición y actitud de vehículos autónomos. Cuando no existe una descripción probabilística de las variables de interés y sólo se conocen las cotas que pueden alcanzar, es preferible utilizar técnicas de estimación determinística basada en conjuntos. Estas técnicas modelan las señales de interés como variables determinísticas acotadas a conjuntos dados. Esto permite utilizar la información disponible al momento del diseño para definir cotas mínimas y máximas sin necesidad de suposiciones restrictivas. En lugar de utilizar propiedades estadísticas para definir la incertidumbre, se utilizan normas para acotarla. De esta forma se pretende lograr una mayor precisión y robustez frente a incertidumbres y perturbaciones para proveer a los vehículos un mayor grado de autonomía, un mejor desempeño

y garantizar su seguridad en condiciones operacionales diversas.

Otra posibilidad es unificar el problema de navegación y guiado en un solo problema de optimización. De esta forma, se combina de forma simultánea la estimación y el control y los algoritmos resultantes serán capaces de considerar que el proceso de estimación influye al de control y el proceso de control influye al de estimación.

Implementar los algoritmos propuestos en computadoras de placa única de bajo costo a fin de que sean utilizada como computadora a bordo para un robot terrestre dotado de sensores tales como GPS e IMU. A tal fin, se debe realizar la fusión de datos entre sensores para estimar la posición y orientación actual del vehículo y a partir de ella, calcular las acciones de control que conduzcan al *waypoint* deseado. Para ello es necesario operar en tiempo real en el orden de los 200 milisegundos.

En el caso de navegar sobre terreno arbitrario con posibles obstáculos, se debe estudiar cómo incorporar esta información a los algoritmos. Para ello se pueden utilizar cámaras de video y flujo óptico, sensores de rango láser (LIDAR), entre otros. A partir de esta información se puede construir un mapa en tiempo real del ambiente que rodea al vehículo y tomar decisiones acordes a la situación en cada instante.

Apéndice A

Demostraciones del Capítulo 3

A.1. Demostración de la Observación 3.3

Del Lema 3.2, P_k es simétrica positiva definida para todo k . Su norma $\|P_k\|$ es igual a $\lambda_{max}(P_k)$, el máximo autovalor de P_k . Su traza satisface $\|P_k\| \leq \text{trace}(P_k)$.

Por elección, $\|P_0\| \leq \text{trace}(P_0) < c$. Suponga que $\|P_k\| \leq c$. De la ecuación (3.13), si $(1/\alpha_k) \text{trace}(W_k) \leq c$, entonces $\|P_k\| = (1/\alpha_k)\|W_k\| \leq (1/\alpha_k) \text{trace}(W_k) \leq c$. Por otro lado, si $(1/\alpha_k) \text{trace}(W_k) > c$, entonces $\alpha_k = 1$ y $P_k^{-1} = P_{k-1}^{-1} + \hat{x}_{k-N|k-1} \hat{x}_{k-N|k-1}^T$, es decir, P_k es *menos* positiva definida que P_{k-1} , por lo tanto $\|P_k\| \leq \|P_{k-1}\| \leq c$. Entonces, $\|P_k\| \leq c$. \square

A.2. Demostración de la Observación 3.4

La fórmula de actualización para P_{k-N} dada por la ecuación (3.13) puede ser vista como un caso especial de una ecuación de Ricatti de la siguiente manera

$$\begin{aligned} \alpha_k P_{k-N} &= P_{k-N-1} - \frac{P_{k-N-1} \hat{x}_{k-N|k-1} \hat{x}_{k-N|k-1}^T P_{k-N-1}}{1 + \hat{x}_{k-N|k-1}^T P_{k-N-1} \hat{x}_{k-N|k-1}} \\ &= P_{k-N-1} - P_{k-N-1} \hat{x}_{k-N|k-1} (1 + \hat{x}_{k-N|k-1}^T P_{k-N-1} \hat{x}_{k-N|k-1})^{-1} \hat{x}_{k-N|k-1}^T P_{k-N-1} \end{aligned}$$

la cual es estable si y solo si todos los autovalores de la matriz de transición a lazo cerrado

$$I - \frac{\hat{x}_{k-N|k-1} \hat{x}_{k-N|k-1}^T P_{k-N-1}}{1 + \hat{x}_{k-N|k-1}^T P_{k-N-1} \hat{x}_{k-N|k-1}} \quad (\text{A.1})$$

están dentro del círculo unitario del plano complejo. Los autovalores del lado derecho de la expresión anterior están dentro del círculo unitario, ya que

$$\|\hat{x}_{k-N|k-1}\hat{x}_{k-N|k-1}^T P_{k-N-1}\| \leq \|\hat{x}_{k-N|k-1}\| \|\hat{x}_{k-N|k-1}^T\| \|P_{k-N-1}\|$$

y

$$\|\hat{x}_{k-N|k-1}^T P_{k-N-1} \hat{x}_{k-N|k-1}\| \leq \|\hat{x}_{k-N|k-1}^T\| \|P_{k-N-1}\| \|\hat{x}_{k-N|k-1}\|$$

entonces

$$\left\| \frac{\hat{x}_{k-N|k-1}\hat{x}_{k-N|k-1}^T P_{k-N-1}}{1 + \hat{x}_{k-N|k-1}^T P_{k-N-1} \hat{x}_{k-N|k-1}} \right\| \leq 1. \quad (\text{A.2})$$

Dado que el producto de matrices simétricas positivas definidas es otra matriz simétrica positiva definida, los autovalores de (A.2) son todos mayores que 0 y menores que 1. Entonces, los autovalores de (A.1) son mayores que 0 y menores que 1. Por lo tanto, la matriz P_k tiene una solución de estado estacionario P_∞ . \square

A.3. Demostración del Teorema 3.1

Dado que $\alpha_k V_{k-1} \leq V_{k-1}$, podemos restar $\alpha_k V_{k-1}$ a V_k

$$\begin{aligned} V_k - \alpha_k V_{k-1} &= \tilde{x}_k^T P_k^{-1} \tilde{x}_k - \alpha_k \tilde{x}_{k-1}^T P_{k-1}^{-1} \tilde{x}_{k-1} \\ &= (A\tilde{x}_{k-1})^T P_k^{-1} (A\tilde{x}_{k-1}) - \alpha_k \tilde{x}_{k-1}^T P_{k-1}^{-1} \tilde{x}_{k-1} \\ &= \tilde{x}_{k-1}^T A^T P_k^{-1} A \tilde{x}_{k-1} - \alpha_k \tilde{x}_{k-1}^T P_{k-1}^{-1} \tilde{x}_{k-1} \\ &= \tilde{x}_{k-1}^T A^T [\alpha_k (P_{k-1}^{-1} + \hat{x}_{k-N|k-1} \hat{x}_{k-N|k-1}^T)] A \tilde{x}_{k-1} - \alpha_k \tilde{x}_{k-1}^T P_{k-1}^{-1} \tilde{x}_{k-1} \\ &= \alpha_k \tilde{x}_{k-1}^T [A^T P_{k-1}^{-1} A + A^T \hat{x}_{k-N|k-1} \hat{x}_{k-N|k-1}^T A - P_{k-1}^{-1}] \tilde{x}_{k-1} \end{aligned}$$

Como P_k es simétrica positiva definida para todo k y $A^T \hat{x}_{k-N|k-1} \hat{x}_{k-N|k-1}^T A$ es simétrica positiva semidefinida para todo k , si el sistema dado por la ecuación (3.15) es observable, entonces MHE con actualización adaptativa del costo de arriba es un observador asintóticamente estable. \square

Apéndice B

Demostraciones del Capítulo 4

B.1. Demostración del Teorema 4.1

Primero se demuestra que la entrada de control y el estado convergen al origen, y luego se demostrará que el origen es un punto de equilibrio estable para el sistema a lazo cerrado. La combinación de convergencia y estabilidad dan como resultado estabilidad asintótica.

Convergencia. La convergencia del estado y las entradas de control al origen puede ser establecida demostrando que la secuencia de valores de la función de costo es no-creciente.

Sea $\mathcal{J}(k)$ la función de costo definida por (4.11), con Q , R y $P_{k|k}$ matrices positivas definidas; $P_{k|k}$ satisface la ecuación de Lyapunov.

Sea $\mathbf{u}_k^* = [u_{k|k}^*, u_{k+1|k}^*, \dots, u_{k+N-1|k}^*]^T$ la secuencia de entradas de control óptima calculada a tiempo k . Si sólo existen restricciones a las entradas de control, entonces la secuencia $\hat{\mathbf{u}}_{k+1} = [u_{k+1|k}^*, u_{k+2|k}^*, \dots, u_{k+N-1|k}^*, 0]^T$ es factible a tiempo $k+1$. Dado que $P_{k|k}$ satisface la ecuación de Lyapunov, entonces la función de costo (4.11) aproxima de forma exacta el problema de control óptimo. Entonces, al evaluar $\mathcal{J}(k)$ para \mathbf{u}_k^* y $\hat{\mathbf{u}}_{k+1}$, y suponiendo que no hay perturbaciones o errores de linealización, se puede demostrar que

$$\hat{\mathcal{J}}(k+1) - \mathcal{J}^*(k) = -x_{k|k}^T Q x_{k|k} - u_{k|k}^* R u_{k|k}^*, \quad (\text{B.1})$$

donde $\hat{\mathcal{J}}(i)$ y $\mathcal{J}^*(i)$ denotan los valores de la función de costo para $\hat{\mathbf{u}}_i$ y \mathbf{u}_i^* , respectivamente. Dado que el lado derecho de (B.1) es definido seminegativo, entonces

$$\hat{\mathcal{J}}(k+1) \leq \mathcal{J}^*(k). \quad (\text{B.2})$$

Pero $\hat{\mathbf{u}}_{k+1}$ es una secuencia factible subóptima, entonces puede afirmarse que

$\mathcal{J}^*(k+1) \leq \tilde{\mathcal{J}}(k+1)$ y en consecuencia

$$\mathcal{J}^*(k+1) \leq \mathcal{J}^*(k) \quad \forall k. \quad (\text{B.3})$$

Esto demuestra que la secuencia de valores de costo óptimos $\{\mathcal{J}^*(k)\}$ decrece sobre trayectorias a lazo cerrado del sistema. El costo está acotado por debajo por cero y de esta forma, tiene un límite no negativo. Por ello, a medida que $k \rightarrow \infty$ la diferencia de costos óptimos $\Delta\mathcal{J}^*(k+1) = \mathcal{J}^*(k+1) - \mathcal{J}^*(k) \rightarrow 0$. Debido a que Q y R son positivos definidos, a medida que $\Delta\mathcal{J}^*(k+1) \rightarrow 0$ los estados y las entradas de control deben converger al origen, $x_k \rightarrow 0$ y $u_k \rightarrow 0$ a medida que $k \rightarrow \infty$.

Estabilidad. Para demostrar que el origen es asintóticamente estable, podemos observar que de (B.3) queda claro que la secuencia de costos óptimos $\{\mathcal{J}^*(k)\}$ es no creciente, lo que implica $\mathcal{J}^*(k) \leq \mathcal{J}^*(0) \forall k > 0$. A tiempo $k = 0$, se puede escribir la función de costo como

$$\mathcal{J}(0) = x_0^T P_0 x_0, \quad (\text{B.4})$$

donde P_0 satisface la ecuación de Lyapunov $P_k - A_k^T P_k A_k = Q$, $Q > 0$. De la definición de función de costo, se puede escribir

$$x_k^T Q x_k \leq \mathcal{J}^*(k), \quad (\text{B.5})$$

entonces,

$$x_k^T Q x_k \leq \mathcal{J}^*(k) \leq \mathcal{J}^*(0) \leq \mathcal{J}(0) = x_0^T P_0 x_0, \quad (\text{B.6})$$

lo que implica

$$x_k^T Q x_k \leq x_0^T P_0 x_0 \quad \forall k. \quad (\text{B.7})$$

Dado que Q and P_0 son definidas positivas, tenemos

$$\lambda_{\min}(Q) \|x_k\|^2 \leq \lambda_{\max}(P_0) \|x_0\|^2 \quad \forall k, \quad (\text{B.8})$$

donde $\lambda_{\min}(\cdot)$ y $\lambda_{\max}(\cdot)$ denotan los autovalores mínimos y máximos de la matriz correspondiente. Finalmente, obtenemos que

$$\|x_k\| \leq \sqrt{\frac{\lambda_{\max}(P_0)}{\lambda_{\min}(Q)}} \|x_0\| \quad \forall k > 0. \quad (\text{B.9})$$

Por lo que el lazo cerrado es estable. La combinación de convergencia y estabilidad implican que el origen es un punto de equilibrio asintóticamente estable del sistema a lazo cerrado. \square

B.2. Demostración del Teorema 4.3

Dado que la función de costo (4.11) es localmente convexa a cada instante de muestreo (por ser una función cuadrática) y dado que solamente se consideran restricciones lineales en las entradas de control, el problema de optimización del Algoritmo 2 es localmente convexo.

Sea \mathbf{u}_k^0 una secuencia factible de entradas de control a tiempo k definida como

$$\mathbf{u}_k^0 = [u_{k|k-1}^*, u_{k+1|k-1}^*, \dots, u_{k+N-2|k-1}^*, 0]^T. \quad (\text{B.10})$$

En la iteración $q = 1$, sea \mathbf{u}_k^1 una combinación convexa factible de \mathbf{u}_k^* y \mathbf{u}_k^0 , es decir

$$\mathbf{u}_k^1 = \alpha \mathbf{u}_k^* + (1 - \alpha) \mathbf{u}_k^0, \quad \text{con } 0 \leq \alpha \leq 1. \quad (\text{B.11})$$

Como la función de costo iterada

$$\mathcal{J}^q(k) = \sum_{i=0}^{N-1} \left[\tilde{x}_{k+i|k}^{qT} Q \tilde{x}_{k+i|k}^q + \tilde{u}_{k+i|k}^{qT} R \tilde{u}_{k+i|k}^q \right] + \tilde{x}_{k+N|k}^{qT} P_{k|k}^q \tilde{x}_{k+N|k}^q, \quad (\text{B.12})$$

es una función localmente convexa, podemos escribir

$$\begin{aligned} \mathcal{J}^1(k) &= \alpha \mathcal{J}^*(k) + (1 - \alpha) \mathcal{J}^0(k), \\ &= \alpha (\mathcal{J}^*(k) - \mathcal{J}^0(k)) + \mathcal{J}^0(k), \end{aligned} \quad (\text{B.13})$$

como $0 \leq \alpha \leq 1$ y $\mathcal{J}^*(k)$ es el valor óptimo de la función de costo a tiempo k , entonces

$$\alpha (\mathcal{J}^*(k) - \mathcal{J}^0(k)) \leq 0, \quad (\text{B.14})$$

en consecuencia

$$\mathcal{J}^1(k) \leq \mathcal{J}^0(k), \quad (\text{B.15})$$

Se puede repetir el mismo argumento en la iteración siguiente para ver que

$$\mathcal{J}^{q+1}(k) \leq \mathcal{J}^q(k), \quad q \geq 0, \quad (\text{B.16})$$

Esto muestra que la secuencia $\{\mathcal{J}^q(k)\}$ es no-creciente. Dado que la función de costo es cuadrática, esta acotada por debajo por cero y tiene un límite no negativo. Entonces, a medida que $q \rightarrow \infty$ la diferencia de las funciones de costo $\Delta \mathcal{J}^q(k) = \mathcal{J}^{q+1} - \mathcal{J}^q \rightarrow 0$, y como resultado $\mathcal{J}^q(k) \rightarrow \mathcal{J}^*(k)$. \square

Apéndice C

Demostraciones del Capítulo 5

C.1. Demostración del Lema 5.1

De la definición de $J(\cdot)$ tenemos

$$\mathcal{J}(x_k, {}^q\mathbf{u}_k, A) = \mathcal{J}\left(x_k, \left[{}^q\mathbf{u}_k^{j \in \mathcal{N}^1} \dots {}^q\mathbf{u}_k^{j \in \mathcal{N}^m} \right], A\right) \quad (\text{C.1})$$

De la definición de ${}^q\mathbf{u}^{j \in \mathcal{N}^l}$, tenemos

$$\begin{aligned} \mathcal{J}(x_k, {}^q\mathbf{u}_k, A) &= \mathcal{J}\left(x_k, \left[\alpha^{j \in \mathcal{N}^1} {}^q\tilde{\mathbf{u}}_k^1 + (1 - \alpha^{j \in \mathcal{N}^1}) {}^{q-1}\mathbf{u}_k^{j \in \mathcal{N}^1} \dots \right. \right. \\ &\quad \left. \left. \alpha^{j \in \mathcal{N}^m} {}^q\tilde{\mathbf{u}}_k^m + (1 - \alpha^{j \in \mathcal{N}^m}) {}^{q-1}\mathbf{u}_k^{j \in \mathcal{N}^m} \right], A\right) \\ &= \mathcal{J}\left(x_k, \left[\alpha^{j \in \mathcal{N}^1} \left[{}^q\tilde{\mathbf{u}}_k^{j \in \mathcal{N}^1} \dots {}^{q-1}\mathbf{u}_k^{j \in \mathcal{N}^m} \right] \dots \right. \right. \\ &\quad \left. \left. \alpha^{j \in \mathcal{N}^m} \left[{}^{q-1}\mathbf{u}_k^{j \in \mathcal{N}^1} \dots {}^q\tilde{\mathbf{u}}_k^{j \in \mathcal{N}^m} \right] \right], A\right) \end{aligned}$$

Por la convexidad de $J(\cdot)$ tenemos

$$\mathcal{J}(x_k, {}^q\mathbf{u}_k, A) \leq \sum_{l=1}^m \alpha^l J\left(x_k, {}^q\tilde{\mathbf{u}}_k^{j \in \mathcal{N}^l}, {}^{q-1}\mathbf{u}_k^{j \in \mathcal{I} - \mathcal{N}^l}, A\right) \quad (\text{C.2})$$

y del Algoritmo 5.2.2 sabemos que

$$\mathcal{J}\left(x_k, {}^q\tilde{\mathbf{u}}_k^{j \in \mathcal{N}^l}, {}^{q-1}\mathbf{u}_k^{j \in \mathcal{I} - \mathcal{N}^l}, A\right) \leq \mathcal{J}\left(x_k, {}^{q-1}\mathbf{u}_k, A\right),$$

entonces

$$\mathcal{J}(x_k, {}^q \mathbf{u}_k, A) \leq \mathcal{J}(x_k, {}^q \tilde{\mathbf{u}}_k^{j \in \mathcal{N}^l}, {}^{q-1} \mathbf{u}_k^{j \in \mathcal{I} - \mathcal{N}^l}, A) \leq \mathcal{J}(x_k, {}^{q-1} \mathbf{u}_k, A). \quad (\text{C.3})$$

Restando las funciones de costo en $q - 1$ y q , obtenemos

$$\Delta \mathcal{J}(x_k, {}^{q-1} \mathbf{u}_k, A) \leq -{}^{q-1} \Delta \mathbf{u}_k^{j \in \mathcal{N}^l T} R {}^{q-1} \Delta \mathbf{u}_k^{j \in \mathcal{N}^l}.$$

Esto muestra que las secuencias de costos $\{\mathcal{J}_k^l\}$ es no creciente y que el costo está acotado por debajo por cero y por lo tanto, tiene un límite no negativo. Por lo tanto, a medida que $q \rightarrow \infty$ la diferencia de costos ${}^q \Delta \mathcal{J}(k) \rightarrow 0$ de forma tal que ${}^q \mathcal{J}(k) \rightarrow \mathcal{J}^*(k)$. como $R > 0$, a medida que ${}^q \Delta \mathcal{J}(k) \rightarrow 0$ las actualizaciones de las entradas ${}^{q-1} \Delta \mathbf{u}_k \rightarrow 0$ as $q \rightarrow \infty$, y la solución del problema de optimización ${}^q \mathbf{u}_k$ converge a una solución $\bar{\mathbf{u}}_k$. Dependiendo de la función de costo utilizada por los controladores distribuidos, $\bar{\mathbf{u}}_k$ puede converger a $\mathbf{u}^*(k)$ (ver la Sección 5.3.1). \square

C.2. Demostración del Teorema 5.1

Primero se demostrará que la entrada y el estado real de la planta convergen al origen y luego se demostrará que el origen es un punto de equilibrio estable para el sistema a lazo cerrado. La combinación de convergencia y estabilidad dan como resultado estabilidad asintótica.

Convergencia. La convergencia de los estados y las entradas al origen pueden determinarse al mostrar que la secuencia de valores de la función de costo es no creciente.

Mostrar estabilidad del sistema a lazo cerrado sigue procedimientos estándar en su mayor parte [79, 92]. En lo que sigue, se describirá solo la parte más importante, que considera la propiedad no creciente del valor de la función de costo. La demostración de esta Sección esta relacionada a la demostración de estabilidad del método *FC-MPC* en [129].

Sean q_k y $q(k + 1)$ el número de iteración del **Algoritmo 1** a tiempo k y $k + 1$ respectivamente. Sean $J(k) = \mathcal{J}(x_k, \mathbf{u}_k, A)$ y $J(k + 1) = \mathcal{J}(x_{k+1}, \mathbf{u}_{k+1}, A)$ los valores de la función de costo asociados con la solución final combinada a tiempo k y $k + 1$. A tiempo $k + 1$, sea $\mathcal{J}^l(k + 1) = \mathcal{J}(x_{k+1}, {}^q \mathbf{u}_k^{j \in \mathcal{N}^l}, {}^{q-1} \mathbf{u}_k^{j \in \mathcal{I} - \mathcal{N}^l}, A)$ el costo global asociado con la solución del subsistema l en la iteración q .

La función de costo global $\mathcal{J}(x_k, \mathbf{u}_k)$ puede ser utilizada como una función de Lyapunov del sistema y su propiedad de no crecimiento puede demostrarse

a continuación

$$\begin{aligned} \mathcal{J}(x_{k+1}, \mathbf{u}_{k+1}, A) &\leq \dots \leq \mathcal{J}(x_{k+1}, {}^q\mathbf{u}(k+1), A) \leq \dots \\ \dots &\leq \mathcal{J}(x_{k+1}, \mathbf{u}^1(k+1), A) \leq \mathcal{J}(x_k, \mathbf{u}_k, A) - x_k^T Q x_k - \mathbf{u}_k^T R \mathbf{u}_k \end{aligned}$$

La desigualdad $\mathcal{J}(x_{k+1}, {}^q\mathbf{u}(k+1), A) \leq \mathcal{J}(x_{k+1}, \mathbf{u}^{q-1+}(k+1), A)$ es consecuencia del Lema 1. Utilizando esta desigualdad, podemos volver hacia atrás hasta $q = 1$

$$\begin{aligned} \mathcal{J}(x_{k+1}, \mathbf{u}_{k+1}, A) &\leq \dots \leq \mathcal{J}(x_{k+1}, {}^q\mathbf{u}(k+1), A) \leq \dots \\ \dots &\leq \mathcal{J}(x_{k+1}, \mathbf{u}^1(k+1), A). \end{aligned}$$

Al tiempo $q = 1$, podemos utilizar la solución inicial factible $\mathbf{u}^0(k+1)$. En esta iteración, el MPC distribuido optimiza la función de costo con respecto a las variables locales comenzando desde $\mathbf{u}^0(k+1)$, por lo tanto $\forall l = 1, \dots, m$

$$\begin{aligned} \mathcal{J}(x_{k+1}, {}^1\mathbf{u}_k^{j \in \mathcal{N}^l}, {}^0\mathbf{u}_k^{j \in \mathcal{I} - \mathcal{N}^l}, A) &\leq \mathcal{J}(x_{k+1}, {}^0\mathbf{u}_k, A) \\ &\leq \sum_{i=1}^{\infty} x_{k+i|k}^T Q x_{k+i|k} + \mathbf{u}_{k+i|k}^T R \mathbf{u}_{k+i|k} \\ &\leq \mathcal{J}(x_k, \mathbf{u}_k, A) - x_k^T Q x_k - \mathbf{u}_k^T R \mathbf{u}_k \end{aligned}$$

Dada la convexidad de J y la actualización utilizando la combinación convexa (Paso 2.c del Algoritmo 1), se obtiene

$$\mathcal{J}(x_k, {}^1\mathbf{u}_k, A) \leq \sum_{l=1}^m \alpha^l \mathcal{J}(x_{k+1}, {}^1\mathbf{u}_k^{j \in \mathcal{N}^l}, {}^0\mathbf{u}_k^{j \in \mathcal{I} - \mathcal{N}^l}, A) \quad (\text{C.4})$$

entonces,

$$\begin{aligned} \mathcal{J}(x_k, {}^1\mathbf{u}_k, A) &\leq \sum_{l=1}^m \alpha^l \left[\mathcal{J}(x_k, \mathbf{u}_k, A) - x_k^T Q x_k - \mathbf{u}_k^T R \mathbf{u}_k \right], \\ &\leq \mathcal{J}(x_k, \mathbf{u}_k, A) - x_k^T Q x_k - \mathbf{u}_k^T R \mathbf{u}_k. \end{aligned}$$

Restando $\mathcal{J}^*(k)$ de $\mathcal{J}^*(k+1)$

$$\mathcal{J}^*(k+1) - \mathcal{J}^*(k) \leq -x_k^T Q x_k - \mathbf{u}_k^T R \mathbf{u}_k \quad \forall k. \quad (\text{C.5})$$

Esto muestra que la secuencia de valores óptimo de la función de costo $\{\mathcal{J}^*(k)\}$ decrece durante las trayectorias a lazo cerrado del sistema. El costo está acotado por debajo por cero y entonces tiene un límite no negativo. Por lo tanto, a medida que $k \rightarrow \infty$ la diferencia de costos óptimos $\Delta \mathcal{J}^*(k+1) \rightarrow 0$.

Como Q y R son matrices positivas definidas, a medida que $\Delta\mathcal{J}^*(k+1) \rightarrow 0$ los estados y las entradas deben converger al origen $x_k \rightarrow 0$ y $\mathbf{u}_k \rightarrow 0$ a medida que $k \rightarrow \infty$.

Escalabilidad. Utilizando la forma QP de (5.8), el costo factible a tiempo $k = 0$ puede ser escrito como sigue: $J(0) = x_0^T \bar{Q} x_0$, donde \bar{Q} es la solución de la función de Lyapunov para la matriz dinámica $\bar{Q} = A^T Q A + Q$.

De la ecuación (C.5) es claro que la secuencia de valores óptimos de la función de costo $\{\mathcal{J}^*(k)\}$ es no creciente, lo que implica $\mathcal{J}^*(k) \leq \mathcal{J}^*(0) \quad \forall k > 0$. De la definición de la función de costo tenemos que $x_k^T Q x_k \leq \mathcal{J}^*(k) \quad \forall k$, lo que implica

$$x_k^T Q x_k \leq x_0^T \bar{Q} x_0 \quad \forall k.$$

Dado que Q y \bar{Q} son positivas definidas, se cumple que

$$\|x_k\| \leq \gamma \|x_0\| \quad \forall k > 0$$

donde

$$\gamma = \sqrt{\frac{\lambda_{\max}(\bar{Q})}{\lambda_{\min}(Q)}}.$$

Por lo tanto, el sistema a lazo cerrado es estable. La combinación de convergencia y estabilidad implican que el origen es un punto de equilibrio asintóticamente estable para el sistema a lazo cerrado. \square

C.3. Demostración del Teorema 5.2

La solución óptima del sistema de control distribuido con fallas de comunicación está dada por

$$\tilde{\mathbf{u}}_k = (I - \mathcal{K}^0 \mathcal{R} \mathcal{C} \mathcal{T})^{-1} \mathcal{K}^1 \mathcal{R} \mathcal{C} \mathcal{T} \Gamma x_k. \quad (\text{C.6})$$

Utilizando la técnica de descomposición de matrices, da como resultado

$$(I - \mathcal{K}^0 \mathcal{R} \mathcal{C} \mathcal{T})^{-1} = (I - \mathcal{K}^0)^{-1} \left(2I - \left[I + (I - \mathcal{K}^0)^{-1} (I + \mathcal{K}^0 - 2\mathcal{K}^0 \mathcal{R} \mathcal{C} \mathcal{T}) \right]^{-1} \right) + (I - \mathcal{K}^0)^{-1}$$

En general $(I - \mathcal{K}^0)^{-1}$ y $(I + \mathcal{K}^0 - 2\mathcal{K}^0 \mathcal{R} \mathcal{C} \mathcal{T})^{-1}$ existen, por lo que la ecuación es válida. De (C.6) tenemos $\mathcal{K}^1 \Gamma x_k = (I - \mathcal{K}^0) \mathbf{u}_k$, entonces $\tilde{\mathbf{u}}_k$ puede ser escrita como una función de la solución óptima \mathbf{u}_k de la siguiente manera

$$\tilde{\mathbf{u}}_k = (\mathcal{S} + I) \mathbf{u}_k$$

donde $\mathcal{S} = 2I - \left[I + (I - \mathcal{K}^0)^{-1} (I + \mathcal{K}^0 - 2\mathcal{K}^0 \mathcal{R} \mathcal{C} \mathcal{T}) \right]^{-1}$.

La función de costo del sistema libre de fallas de comunicación \mathcal{J}^* puede ser escrita en función de \mathbf{u}_k de la siguiente manera

$$\mathcal{J}^* = \left\| (\mathcal{K}^1)^{-1} (I - \mathcal{K}^0) \mathbf{u}_k - \mathcal{H} \mathbf{u}_k \right\|_Q^2 + \left\| \mathbf{u}_k \right\|_R^2 = \left\| \mathbf{u}_k \right\|_{\mathcal{F}}^2 \quad (\text{C.7})$$

donde

$$\mathcal{F} = \left((\mathcal{K}^1)^{-1} (I - \mathcal{K}^0) - \mathcal{H} \right)^T Q \left((\mathcal{K}^1)^{-1} (I - \mathcal{K}^0) - \mathcal{H} \right) + R.$$

En el caso del sistema con de fallas de comunicación, tenemos

$$\tilde{\mathcal{J}} \leq \mathcal{J}^* + \left\| \mathbf{u}_k \right\|_{\mathcal{W}}^2 \quad (\text{C.8})$$

donde $\mathcal{W} = \mathcal{S}^T (\mathcal{H}^T Q \mathcal{H} + R) \mathcal{S}$. Finalmente, el efecto de la comunicación puede relacionarse con \mathcal{J}^* a través de

$$\left\| \mathbf{u}_k \right\|_{\mathcal{W}}^2 \leq \frac{\|\mathcal{W}\|}{\lambda_{\min}(\mathcal{F})} \mathcal{J}^*, \quad (\text{C.9})$$

donde λ_{\min} denota el autovalor mínimo de \mathcal{F} . De las derivaciones anteriores, la relación entre $\tilde{\mathcal{J}}$ y \mathcal{J}^* está dada por

$$\tilde{\mathcal{J}} \leq \left(1 + \frac{\|\mathcal{W}\|}{\lambda_{\min}(\mathcal{F})} \right) \mathcal{J}^*. \quad (\text{C.10})$$

y la degradación es

$$\frac{\tilde{\mathcal{J}} - \mathcal{J}^*}{\mathcal{J}^*} \leq \frac{\|\mathcal{W}\|}{\lambda_{\min}(\mathcal{F})}. \quad (\text{C.11})$$

La inspección de (C.8) muestra que $\|\mathcal{W}\|$ depende de \mathcal{R} y \mathcal{T} . Entonces, en el caso de que existan todas las fallas de comunicación, $\|\mathcal{W}\|$ puede alcanzar su valor máximo

$$\mathcal{W}_{\max} = \left(2I - \left[I + (I - \mathcal{K}^0)^{-1} (I + \mathcal{K}^0) \right]^{-1} \right)^T \left(\mathcal{H}^T Q \mathcal{H} + R \right) \left(2I - \left[I + (I - \mathcal{K}^0)^{-1} (I + \mathcal{K}^0) \right]^{-1} \right),$$

y la cota superior de la desviación del desempeño es

$$\frac{\tilde{\mathcal{J}} - \mathcal{J}^*}{\mathcal{J}^*} \leq \frac{\|\mathcal{W}_{\max}\|}{\lambda_{\min}(\mathcal{F})}. \quad (\text{C.12})$$

□

Siglas

2D dos dimensiones. 21, 23, 43, 105

3D tres dimensiones. 22, 23, 33, 105

DAE Differential Algebraic Equation. 23

DCM Direction Cosine Matrix. 27–29, 31, 32, 36

DH Denavit–Hartenbert. 23

DOF Degrees of Freedom. 23

EKF Extended Kalman Filter. 49, 50, 68, 69

FIE Full Information Estimator. 62–65, 68, 70, 71

GNSS Global Navigation Satellite System. 49, 50

IMU Inertial Measurement Unit. 49, 50

LTI Linear Time Invariant. 88

LTV Linear Time Varying. 77, 78, 81, 82

MHE Moving Horizon Estimation. 49, 50, 55, 59, 62–65, 68, 70, 71, 74, 91, 105

MPC Model Predictive Control. 75, 76, 83, 87–89, 96–99, 103, 117

NLP Non-linear Program. 75, 106

NLMPC Non-linear Model Predictive Control. 75, 76, 78, 81, 84–86, 106

PF Particle Filter. 49

QP Quadratic Programming. 76, 100, 106, 118

RGAS Robust Global Asymptotically Stable. 63

RLS Recursive Least Squares. 59, 60

UKF Unscented Kalman Filter. 49, 69

WMR Wheeled Mobile Robot. 21–24, 37, 43, 44, 105

Bibliografía

- [1] M. Abdolhosseini, Y. Zhang, and C. Rabbath. An efficient model predictive control scheme for an unmanned quadrotor helicopter. *Journal of Intelligent & Robotic Systems*, 70(1-4):27–38, 2013.
- [2] N. Aguilera and J. Marchetti. Optimizing and controlling the operation of heat exchanger networks. *AIChE Journal*, 44(5):1090–1104, 1998.
- [3] A. Al-Matouq and T. Vincent. Multiple window moving horizon estimation. *Automatica*, 53:264–274, 2015.
- [4] K. Alexis, G. Nikolakopoulos, and A. Tzes. Model predictive quadrotor control: attitude, altitude and position experimental studies. *Control Theory Applications, IET*, 6(12):1812–1827, Aug 2012.
- [5] J. Andersson. *A General-Purpose Software Framework for Dynamic Optimization*. PhD thesis, Arenberg Doctoral School, KU Leuven, Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium, October 2013.
- [6] Appin Knowledge Solutions. *Robotics*. Jones & Bartlett Learning, 2010.
- [7] E. Aske, S. Strand, and S. Skogestad. Coordinator mpc for maximizing plant throughput. *Computers and Chemical Engineering*, 32(1-2):195–204, 2008.
- [8] K. Åström and B. Wittenmark. On self tuning regulators. *Automatica*, 9(2):185–199, 1973.
- [9] E. Bekir. *Introduction to Modern Navigation Systems*. World Scientific Publishing Co. Pte. Ltd., 2007.

-
- [10] A. Bemporad, C. Filippi, and F. Torrisi. Inner and outer approximations of polytopes using boxes. *Computational Geometry: Theory and Applications*, 27(2):151–178, 2004.
- [11] L. Biegler, X. Yang, and G. Fischer. Advances in sensitivity-based nonlinear model predictive control and dynamic real-time optimization. *Journal of Process Control*, 30:104 – 116, 2015.
- [12] A. Brooms and B. Kouvaritakis. Successive constrained optimization and interpolation in non-linear model based predictive control. *International journal of control*, 73(4):312–316, 2000.
- [13] E. Camponogara, D. Jia, B. Krogh, and S. Talukdar. Distributed model predictive control. *IEEE Control Systems Magazine*, 22(1):44–52, 2002.
- [14] M. Cannon, J. Buerger, B. Kouvaritakis, and S. Rakovic. Robust Tubes in Nonlinear Model Predictive Control. *IEEE Transactions on Automatic Control*, 56(8):1942–1947, 2011.
- [15] A. Carmi and Y. Oshman. Adaptive Particle Filtering for Spacecraft Attitude Estimation from Vector Observations. *Journal of Guidance, Control, and Dynamics*, 32(1):232–241, jan 2009.
- [16] R. Cheng, J. Forbes, and W. Yip. Price-driven coordination method for solving plant-wide mpc problems. *Journal of Process Control*, 17(5):429–438, 2007.
- [17] R. Cheng, J. Fraser Forbes, and W. Yip. Dantzig–wolfe decomposition and plant-wide mpc coordination. *Computers and Chemical Engineering*, 32(7):1507–1522, 2008.
- [18] Y. Cheng and J. Crassidis. Particle Filtering for Sequential Spacecraft Attitude Estimation. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Reston, Virginia, aug 2004. American Institute of Aeronautics and Astronautics.
- [19] E. Chu, A. Keshavarz, D. Gorinevsky, and S. Boyd. Moving horizon estimation for staged qp problems. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 3177–3182. IEEE, 2012.
- [20] I. Cox and G. Wilfong, editors. *Autonomous Robot Vehicles*. Springer New York, New York, NY, 1990.
- [21] J. Crassidis and J. Junkins. *Optimal estimation of dynamic systems*. CRC press, 2011.

- [22] J. Crassidis and F. Markley. Unscented Filtering for Spacecraft Attitude Estimation. *Journal of Guidance, Control, and Dynamics*, 26(4):536–542, jul 2003.
- [23] J. Cueli and C. Bordons. Iterative nonlinear model predictive control. Stability, robustness and applications. *Control Engineering Practice*, 16(9):1023–1034, 2008.
- [24] F. Cuzzola, J. Geromel, and M. Morari. An improved approach for constrained robust model predictive control. *Automatica*, 38(7):1183 – 1189, 2002.
- [25] N. De Oliveira. *Newton-type algorithms for nonlinear constrained chemical process control*. PhD thesis, Carnegie Mellon University, 1994.
- [26] N. De Oliveira and L. Biegler. An extension of newton-type algorithms for nonlinear process control. *Automatica*, 31(2):281–286, 1995.
- [27] S. De Oliveira Kothare and M. Morari. Contractive model predictive control for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 45(6):1053–1071, 2000.
- [28] J. Diebel. Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors. 2006.
- [29] M. Diehl, H. Bock, H. Diedam, and P. Wieber. Fast Direct Multiple Shooting Algorithms for Optimal Robot Control. In *Fast Motions in Biomechanics and Robotics*, pages 65–93. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [30] M. Diehl, H. Ferreau, and N. Haverbeke. Efficient numerical methods for nonlinear mpc and moving horizon estimation. *Nonlinear Model Predictive Control*, pages 391–417, 2009.
- [31] M. Diehl, H. Ferreau, and N. Haverbeke. *Nonlinear Model Predictive Control: Towards New Challenging Applications*, chapter Efficient, pages 391–417. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [32] P. Dubey and J. Rogawski. Inefficiency of smooth market mechanisms. *Journal of Mathematical Economics*, 19(3):285–304, 1990.
- [33] G. Dudek and M. Jenkin. *Computational principles of mobile robotics*. Cambridge university press, 2010.

-
- [34] W. Dunbar. Distributed receding horizon control of dynamically coupled nonlinear systems. *IEEE Transactions on Automatic Control*, 52(7):1249–1263, 2007.
- [35] W. Dunbar and R. Murray. Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42(4):549–558, 2006.
- [36] P. Falcone, F. Borrelli, H. Tseng, J. Asgari, and D. Hrovat. Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation. *International journal of robust and nonlinear control*, 18(8):862–875, 2007.
- [37] P. Falcone, M. Tufo, F. Borrelli, J. Asgari, and H. Tseng. A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems. In *Decision and Control, 2007 46th IEEE Conference on*, pages 2980–2985. IEEE, 2007.
- [38] P. Findeisen. Moving horizon state estimation of discrete time systems. Master’s thesis, University of Wisconsin, Madison, 1997.
- [39] T. Fortescue, L. Kershenbaum, and B. Ydstie. Implementation of self-tuning regulators with variable forgetting factors. *Automatica*, 17(6):831–835, 1981.
- [40] L. Giovanini, J. Balderud, and R. Katebi. Autonomous and decentralized mission planning for clusters of UUVs. *International Journal of Control*, 80(7):1169–1179, 2007.
- [41] H. Goldstein, C. Poole, and J. Safko. *Classical Mechanics*, 2007.
- [42] G. Goodwin, M. Salgado, and E. Silva. Time-domain performance limitations arising from decentralized architectures and their relationship to the rga. *International Journal of Control*, 78(13):1045–1062, 2005.
- [43] Y. Haimes and V. Chankong. *Multiobjective decision making: Theory and methodology*. North Holland, New York, 1983.
- [44] D. Hall and S. McMullen. *Mathematical techniques in multisensor data fusion*. Artech House, 2004.
- [45] E. Haseltine and J. Rawlings. Critical evaluation of extended kalman filtering and moving-horizon estimation. *Industrial & engineering chemistry research*, 44(8):2451–2460, 2005.
- [46] T. Haslwante. *3D Kinematics*. PhD thesis, 2015.

- [47] M. Hovd and S. Skogestad. Sequential design of decentralized controllers. *Automatica*, 30:1601–1601, 1994.
- [48] D. Hsu. Comparison of four gravity models. In *Proceedings of Position, Location and Navigation Symposium - PLANS '96*, pages 631–635. IEEE.
- [49] J. Z. J. Sanz-Subirana and M. Hernández-Pajares. Transformations between ECEF and ENU coordinates. http://www.navipedia.net/index.php/Transformations_between_ECEF_and_ENU_coordinates, 2011. [Online; accessed 30-11-2016].
- [50] M. Jamoom, E. Feron, and M. McConley. Optimal distributed actuator control grouping schemes. In *Proceedings of the 37th IEEE Conference on Decision and Control*, volume 2, pages 1900–1905, 2002.
- [51] A. Jazwinski. *Stochastic Processes and Filtering Theory*, volume 64. Academic Press, 1970.
- [52] D. Jia and B. Krogh. Distributed model predictive control. In *American Control Conference, 2001. Proceedings of the 2001*, volume 4, 2001.
- [53] D. Jia and B. Krogh. Min-max feedback model predictive control for distributed control with communication. In *American Control Conference, 2002. Proceedings of the 2002*, volume 6, 2002.
- [54] C. Johnson. Positive Definite Matrices. *The American Mathematical Monthly*, 77(3):259, mar 1970.
- [55] A. Kelly. *Mobile Robotics: Mathematics, Models, and Methods*. Cambridge University Press, 2013.
- [56] A. Kelly and N. Seegmiller. A Vector Algebra Formulation of Mobile Robot Velocity Kinematics. pages 613–627. 2014.
- [57] A. Kelly and N. Seegmiller. Recursive kinematic propagation for wheeled mobile robots. *The International Journal of Robotics Research*, 34(3):288–313, mar 2015.
- [58] S. Ko and R. Bitmead. State estimation for linear systems with state equality constraints. *Automatica*, 43(8):1363–1368, 2007.
- [59] N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2149–2154. IEEE, 2004.

-
- [60] P. Kühnl, M. Diehl, T. Kraus, J. Schlöder, and H. Bock. A real-time algorithm for moving horizon state and parameter estimation. *Computers & chemical engineering*, 35(1):71–83, 2011.
- [61] A. Kumar and P. Daoutidis. Nonlinear dynamics and control of process systems with recycle. *Journal of Process Control*, 12(4):475–484, 2002.
- [62] I. Landau, R. Lozano, M. M’Saad, and A. Karimi. *Adaptive control*, volume 51. Springer Berlin, 1998.
- [63] W. Langson, I. Chrysochoos, S. Raković, and D. Mayne. Robust model predictive control using tubes. *Automatica*, 40(1):125–133, 2004.
- [64] S. LaValle. *Planning algorithms*. Cambridge University Press, 2006.
- [65] Y. Lee, B. Kouvaritakis, and M. Cannon. Constrained receding horizon predictive control for nonlinear systems. *Automatica*, 38(12):2093–2102, 2002.
- [66] E. Lefferts, F. Markley, and M. Shuster. Kalman filtering for spacecraft attitude estimation, 1982.
- [67] W. Li and L. Biegler. A multistep, newton-type control strategy for constrained, nonlinear processes. In *American Control Conference, 1989*, pages 1526–1527. IEEE, 1989.
- [68] L. Ljung and S. Gunnarsson. Adaptation and tracking in system identification—A survey. *Automatica*, 26(1):7–21, jan 1990.
- [69] R. Lopez-Negrete, F. D’Amato, L. Biegler, and A. Kumar. Fast nonlinear model predictive control: Formulation and industrial process applications. *Computers & Chemical Engineering*, 51:55 – 64, 2013.
- [70] R. López-Negrete, S. Patwardhan, and L. Biegler. Constrained particle filter approach to approximate the arrival cost in moving horizon estimation. *Journal of Process Control*, 21(6):909–919, 2011.
- [71] J. Lu. Challenging control problems and emerging technologies in enterprise optimization. *Control Engineering Practice*, 11(8):847–858, 2003.
- [72] J. Maciejowski. *Predictive control: with constraints*. Prentice Hall, 2002.
- [73] A. Marasco, S. Givigi, and C. Rabbath. Model predictive control for the dynamic encirclement of a target. In *American Control Conference (ACC), 2012*, pages 2004–2009, June 2012.

- [74] J. Marden. *Learning in Large-Scale Games and Cooperative Control*. PhD thesis, University of California, Los Angeles, California, USA, 2007.
- [75] J. Marden, G. Arslan, and J. Shamma. Cooperative control and potential games. *Trans. Sys. Man Cyber. Part B*, 39(6):1393–1407, 2009.
- [76] F. Markley and J. Sedlak. Kalman Filter for Spinning Spacecraft Attitude Estimation. *Journal of Guidance, Control, and Dynamics*, 31(6):1750–1760, nov 2008.
- [77] J. Martínez. *Model order reduction of nonlinear dynamic systems using multiple projection bases and optimized state-space sampling*. PhD thesis, University of Pittsburgh, 2009.
- [78] D. Mayne, E. Kerrigan, E. Van Wyk, and P. Falugi. Tube-based robust nonlinear model predictive control. *International Journal of Robust and Nonlinear Control*, 21(11):1341–1353, 2011.
- [79] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.
- [80] M. Morari and J. Lee. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4-5):667–682, 1999.
- [81] N. Motee and B. Sayyar-Rodsari. Optimal partitioning in distributed model predictive control. In *Proceedings of the American Control Conference*, volume 6, pages 5300–5305, 2003.
- [82] P. Muir and C. Neuman. Kinematic modeling of wheeled mobile robots. *Journal of Robotic Systems*, 4(2):281–340, apr 1987.
- [83] M. Muller. Nonlinear moving horizon estimation for systems with bounded disturbances. In *2016 American Control Conference (ACC)*, pages 883–888. IEEE, jul 2016.
- [84] M. Murillo, G. Sánchez, and L. Giovanini. Iterated non-linear model predictive control based on tubes and contractive constraints. *{ISA} Transactions*, 62:120–128, 2016.
- [85] K. Muske, J. Rawlings, and J. Lee. Receding horizon recursive state estimation. In *American control conference*, number 30, pages 900–904, 1993.

-
- [86] J. Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.
- [87] R. Neck and E. Dockner. Conflict and cooperation in a model of stabilization policies: a differential game approach. *J. Econ. Dyn. Cont.*, 11:153–158, 1987.
- [88] M. Osborne and A. Rubinstein. *A course in game theory*. MIT press, 1994.
- [89] A. Osorio Cordero and D. Mayne. Deterministic convergence of a self-tuning regulator with variable forgetting factor. In *Control Theory and Applications, IEE Proceedings D*, volume 128, pages 19–23. IET, 1981.
- [90] S. Patwardhan, S. Narasimhan, P. Jagadeesan, B. Gopaluni, and S. Shah. Nonlinear bayesian state estimation: A review of recent developments. *Control Engineering Practice*, 20(10):933–953, 2012.
- [91] T. Polóni, B. Rohal-Ilkiv, and T. Johansen. Moving Horizon Estimation for Integrated Navigation Filtering. *IFAC-PapersOnLine*, 48(23):519–526, 2015.
- [92] J. Primbs and V. Nevistic. Feasibility and stability of constrained finite receding horizon control. *Automatica*, 36(7):965–971, 2000.
- [93] C. Qu and J. Hahn. Computation of arrival cost for moving horizon estimation via unscented kalman filtering. *Journal of Process Control*, 19(2):358–363, 2009.
- [94] S. Rakovic and D. Mayne. Set robust control invariance for linear discrete time systems. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pages 975–980, Dec 2005.
- [95] C. Rao. *Moving horizon strategies for the constrained monitoring and control of nonlinear discrete-time systems*. PhD thesis, University of Wisconsin, Madison, 2000.
- [96] C. Rao and J. Rawlings. Constrained process monitoring: Moving-horizon approach. *AIChE journal*, 48(1):97–109, 2002.
- [97] C. Rao, J. Rawlings, and J. Lee. Constrained linear state estimation—a moving horizon approach. *Automatica*, 37(10):1619–1628, 2001.

- [98] C. Rao, J. Rawlings, and D. Mayne. Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations. *Automatic Control, IEEE Transactions on*, 48(2):246–258, 2003.
- [99] J. Rawlings and D. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Pub., 2009.
- [100] M. Rhudy, Y. Gu, J. Gross, S. Gururajan, and M. R. Napolitano. Sensitivity Analysis of Extended and Unscented Kalman Filters for Attitude Estimation. *Journal of Aerospace Information Systems*, 10(3):131–143, mar 2013.
- [101] M. Risbeck, N. Patel, and J. Rawlings. Nonlinear model predictive control tools for casadi. <https://bitbucket.org/rawlings-group/mpc-tools-casadi>, 2015. Accessed: 2016-11-10.
- [102] D. Robertson and J. Lee. On the use of constraints in least squares estimation and control. *Automatica*, 38(7):1113–1123, 2002.
- [103] J. Rossiter. *Model-based predictive control: a practical approach*. CRC press, 2003.
- [104] S. Roumeliotis and G. Bekey. 3-D Localization for a Mars Rover Prototype. In *Artificial Intelligence, Robotics and Automation in Space, Proceedings of the Fifth International Symposium, ISAIRAS '99*, pages 441–448, 1999.
- [105] M. Salgado and A. Conley. MIMO interaction measure and controller structure selection. *International Journal of Control*, 77(4):367–383, 2004.
- [106] G. Sanchez, A. Limache, L. Giovanini, and M. Murillo. Distributed model predictive control based on dynamic games. In T. Zheng, editor, *Advanced Model Predictive Control*, chapter 4, pages 65–90. InTech, 7 2011.
- [107] N. Sandell Jr, P. Varaiya, M. Athans, and M. Safonov. Survey of decentralized control methods for large scale systems. *IEEE Transactions on Automatic Control*, 23(2):108–128, 1978.
- [108] C. Schumacher, P. Chandler, and S. Rasmussen. Task allocation for wide area search munitions. In *American Control conference*, pages 1917–1922, Anchorage, Alaska, 2002.

-
- [109] N. Seegmiller. *Dynamic Model Formulation and Calibration for Wheeled Mobile Robots*. Phd thesis, Carnegie Mellon University, 2014.
- [110] N. Seegmiller and A. Kelly. Enhanced 3D Kinematic Modeling of Wheeled Mobile Robots. In *Robotics: Science and Systems X*. Robotics: Science and Systems Foundation, jul 2014.
- [111] N. Seegmiller and A. Kelly. High-Fidelity Yet Fast Dynamic Models of Wheeled Mobile Robots. *IEEE Transactions on Robotics*, 32(3):614–625, jun 2016.
- [112] T. Shima, S. Rasmussen, and A. Sparks. Uav cooperative multiple task assignments using genetic algorithms. In *American Control Conference*, pages 2989–2994, Portland, Oregon, USA, 2005.
- [113] B. Siciliano and O. Khatib. *Springer handbook of robotics*. Springer Science & Business Media, 2008.
- [114] R. Siegwart, I. Nourbakhsh, and D. Scaramuzza. *Introduction to autonomous mobile robots*. MIT press, 2011.
- [115] D. Siljak. Decentralized control and computations: status and prospects. *Annual Reviews in Control*, 20:131–141, 1996.
- [116] D. Simon. Kalman filtering with state constraints: a survey of linear and nonlinear algorithms. *Control Theory & Applications, IET*, 4(8):1303–1318, 2010.
- [117] D. Simon and T. Chia. Kalman filtering with state equality constraints. *Aerospace and Electronic Systems, IEEE Transactions on*, 38(1):128–136, 2002.
- [118] D. Simon and D. Simon. Constrained kalman filtering via density function truncation for turbofan engine health estimation. *International Journal of Systems Science*, 41(2):159–171, 2010.
- [119] M. Tarokh and G. McDermott. Kinematics modeling and analyses of articulated rovers. *IEEE Transactions on Robotics*, 21(4):539–553, aug 2005.
- [120] M. Tarokh and G. McDermott. A Systematic Approach to Kinematics Modeling of High Mobility Wheeled Rovers. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 4905–4910. IEEE, apr 2007.

- [121] B. Teixeira, J. Chandrasekar, H. Palanthandalam-Madapusi, L. Tôrres, L. Aguirre, and D. Bernstein. Gain-constrained kalman filtering for linear and nonlinear systems. *Signal Processing, IEEE Transactions on*, 56(9):4113–4123, 2008.
- [122] M. Tenny and J. Rawlings. Efficient moving horizon estimation and nonlinear model predictive control. In *American Control Conference, 2002. Proceedings of the 2002*, volume 6, pages 4475–4480. IEEE, 2002.
- [123] D. Titterton and J. Weston. *Strapdown Inertial Navigation Technology*, volume 17. 2004.
- [124] S. Ungarala. Computing arrival cost parameters in moving horizon estimation using sampling based filters. *Journal of Process Control*, 19(9):1576–1588, 2009.
- [125] M. Vaccarini, S. Longhi, and M. Katebi. Unconstrained networked decentralized model predictive control. *Journal of Process Control*, 19(2):328–339, 2009.
- [126] E. Van Henten and J. Bontsema. Time-scale decomposition of an optimal control problem in greenhouse climate management. *Control Engineering Practice*, 17(1):88–96, 2009.
- [127] J. Vandersteen, M. Diehl, C. Aerts, and J. Swevers. Spacecraft Attitude Estimation and Sensor Calibration Using Moving Horizon Estimation. *Journal of Guidance, Control, and Dynamics*, 36(3):734–742, may 2013.
- [128] S. Vazquez, J. Leon, L. Franquelo, J. Rodriguez, H. Young, A. Marquez, and P. Zanchetta. Model predictive control: A review of its applications in power electronics. *Industrial Electronics Magazine, IEEE*, 8(1):16–31, March 2014.
- [129] A. Venkat, I. Hiskens, J. Rawlings, and S. Wright. Distributed mpc strategies with application to power system automatic generation control. *IEEE Transactions on Control Systems Technology*, 16(6):1192–1206, 2008.
- [130] K. Waldron and J. Schmiedeler. Kinematics. *Springer handbook of robotics*, pages 9–33, 2008.
- [131] B. Wittenmark and M. Salgado. Hankel-norm based interaction measure for input-output pairing. In *Proc. of the 2002 IFAC World Congress*, 2002.

- [132] T. Yang and E. Polak. Moving horizon control of nonlinear systems with input saturation, disturbances and plant uncertainty. *International Journal of Control*, 58(4):875–903, 1993.
- [133] S. Yu, C. Maier, H. Chen, and F. Allgöwer. Tube mpc scheme based on robust control invariant set with application to lipschitz nonlinear systems. *Systems and Control Letters*, 62(2):194–200, 2013.
- [134] Y. Zhang and S. Li. Networked model predictive control based on neighbourhood optimization for serially connected large-scale processes. *Journal of process control*, 17(1):37–50, 2007.
- [135] G. Zhu and M. Henson. Model predictive control of interconnected linear and nonlinear processes. *Industrial and Engineering Chemistry Research*, 41(4):801–816, 2002.

Doctorado en Ingeniería
Mención Inteligencia Computacional, Señales y Sistemas

Título de la obra:

**Algoritmos de Navegación y Guiado
para Vehículos Autónomos
Basados en Métodos de Horizonte Móvil**

Autor: Guido Marcelo Sánchez

Lugar: Santa Fe, Argentina

Palabras Clave:

Métodos de horizonte móvil,
Estimación con restricciones,
Control con restricciones,
Sistemas multiagente,
Teoría de juegos.