"SISTEMA DE POSICIONAMIENTO ABSOLUTO DE UN ROBOT MÓVIL FUSIÓN DE SENSADO ODOMÉTRICO Y VISIÓN DE MARCAS ARTIFICIALES"

Trabajo de Investigación:

PROCESO DE INFORMACIÓN PARA MEDIDA DE DISTANCIAS Y POSICIONAMIENTO 3D DE MÓVILES UTILIZANDO VISIÓN ARTIFICIAL

Tutor: Dr. D. Manual Mazo Quintas
Autora: Marta Marrón Romera

Índice

1. Introducción

En este documento se expone el trabajo titulado “Sistema de posicionamiento absoluto de un robot móvil. Fusión de sensado odométrico y visión de marcas artificiales”, dentro del área de investigación “Proceso de Información para Medida de Distancias y Posicionamiento 3D de Móviles utilizando Visión Artificial” del programa de doctorado “Electrónica” del Departamento de Electrónica de la Universidad de Alcalá, dirigido por Dr. D. Manual Mazo Quintas, y con la colaboración de los Drs. D. Juan Carlos García García y D. Miguel Ángel Sotelo Vázquez.

Dentro del área de investigación elegida, y tal y como muestra el título concreto de esta tesina, el trabajo se ha centrado en el estudio, diseño y aplicación de técnicas de fusión sensorial para el posicionamiento de robots móviles. Más concretamente, el trabajo pretende dar un paso más en varias líneas de investigación abordadas previamente en el Departamento de Electrónica. Por una parte retoma la línea desarrollada por el Dr. D. Juan Carlos García García en su tesis doctoral [García-01] de diseño de algoritmos de posicionamiento absoluto (SPL) para robots móviles. Dichos algoritmos, que serán sólo parcialmente revisados en este trabajo, permiten obtener la posición y orientación relativa del móvil ($\vec{z}_v' = \begin{bmatrix} x_v' & y_v' & \theta_v' \end{bmatrix}^T$)[1], con una incertidumbre bien caracterizada, con respecto a una serie de marcas artificiales colocadas estratégicamente en el entorno de movimiento. A partir de la información obtenida por este algoritmo y con la ayuda de un mapa del entorno global se puede obtener la posición absoluta de robot $\vec{z}_v = \begin{bmatrix} x_v & y_v & \theta_v \end{bmatrix}^T$, como se comentará más adelante en la memoria. Los experimentos prácticos emplearon como plataforma de aplicación una silla de ruedas eléctrica automatizada [Sebastián-99].

La información obtenida por el sistema de visión puede ser utilizada para implementar tareas de navegación autónoma o semi-autónoma del robot en ciertos entornos. Este objetivo ya fue abordado por la

[1] A lo largo de todo el documento se utilizará la notación subíndice ‘o’ para indicar que la información se obtiene por odometría, mientras que el subíndice ‘v’ para indicar que ésta es obtenida mediante el sistema de visión. Esta notación se obviará en el capítulo segundo del documento y se retomará en el de modelado del sistema.

autora de este trabajo de investigación, que desarrolló un sistema de navegación autónoma por entornos interiores parcialmente estructurados [Marrón-00] para la misma silla de ruedas automatizada. En dicho trabajo, la navegación se lleva a cabo mediante una arquitectura jerárquica de procesamiento y en base a un mapa de nodos del entorno muy sencillo, obtenido off-line, que hacía uso de las características de estructuración del medio. La información de posición que manejaba este navegador era simplemente odométrica ($\vec{x}_o = \begin{bmatrix} x_o & y_o & \theta_o \end{bmatrix}^T$), lo cual hizo casi inviable obtener resultados prácticos de recorridos más o menos complejos.

Con estas premisas, el objetivo del trabajo consiste en fusionar los datos de posicionamiento absoluto obtenidos por el sistema de visión con la información de odometría proporcionada por el sistema de encoders del robot móvil (en este caso la silla de ruedas) para poder realizar, a partir de esta fusión, tareas de navegación complejas. Tras revisar diferentes trabajos de fusión realizados por otros grupos de investigación, se observa que la técnica del Filtro de Kalman Extendido (EKF) es la más extendida y adecuada para la aplicación de interés (sistemas no lineales con ruido blanco y gaussiano y con un modelo en variables de estado conocido y fácilmente linealizable alrededor del punto de trabajo). Es por ello que el trabajo de investigación desarrollado se centra en el uso de este estimador óptimo para fusionar los datos procedentes de los sistemas sensoriales comentados, si bien otras técnicas más o menos relacionadas que han sido también analizadas se comentan del mismo modo en la memoria.

El documento que aquí se presenta pretende resumir todos los aspectos estudiados relacionados con el tema, para lo cuál se distinguen los siguientes apartados:

En el punto 2 se realiza una revisión de distintos trabajos de fusión desarrollados en el mundo científico de la robótica móvil, de modo que se pueda obtener un punto de vista objetivo sobre el planteamiento más adecuado en el desarrollo de los procesos de fusión. Se analizan distintas técnicas de fusión y se observa si existe alguna más adecuada para la aplicación concreta que aquí interesa, concluyendo con la elección de un EKF (técnica distribuida y Bayesiana para sistemas no lineales que soportan ruido blanco y gaussiano) como algoritmo más adecuado para este caso. De este modo se establecen las premisas necesarias para implementar este tipo de estimador óptimo en el posicionamiento absoluto de la silla de ruedas.

En el punto siguiente se analiza en detalle el algoritmo del EKF desde un punto de vista general (sin aplicarlo aún al modelo concreto del sistema móvil bajo estudio). Así, se desarrolla inicialmente la algoritmia del Filtro de Kalman (KF) básico para sistemas lineales y luego se adapta a sistemas no lineales, para obtener la del EKF. Además, en este punto se obtienen las conclusiones pertinentes respecto a las características del sistema estocástico a fusionar.

A continuación, en el 4º punto, se realiza el estudio específico del modelo del sistema de interés. Se eligen las variables de estado del sistema, que vendrán dadas en este caso por la cinemática dead-reckoning del robot, y las variables de salida del sistema, que serán en este caso las medidas de posición y orientación obtenidas por el sistema de visión. En este punto, por tanto, se obtienen las matrices y ecuaciones que identifican al sistema bajo estudio, y al ruido asociado a las medidas de estado (odometría) y salida (visión) (ver ecuación <1.1>).

$$\begin{aligned} \vec{x}_{o,k} &= f\left(\vec{x}_{o,k-1}, \vec{u}_k, \vec{w}_k\right) \Rightarrow \; de\, dead-reckoning \\ \vec{z}_{v,k} &= f\left(\vec{x}_{v,k-1}, \vec{v}_k\right) \quad \Rightarrow \quad de\, vision \end{aligned} \qquad <1.1>$$

Con todo ello, en el siguiente punto, se analiza la aplicación del algoritmo EKF, al modelo del sistema que se ha obtenido en el punto 3º. De este modo se obtiene el algoritmo de fusión concreto para el sistema de la silla de ruedas. Además en este apartado final se realizan las consideraciones oportunas sobre tiempos de ejecución (hay que observar que el periodo de obtención de las salidas de visión no coincide con el de evolución del sistema dead-reckoning) y se observa el comportamiento del algoritmo de fusión específicamente diseñado para este sistema ante diversas situaciones.

Para finalizar se incluye un apartado sobre conclusiones del trabajo, y consideraciones a tener en cuenta en la implementación on-line del mismo. Además se establecen líneas de trabajo que aparecen a raíz del estudio y que podrían ser objeto de posteriores trabajos de investigación.

A lo largo de todo el documento se muestran resultados y simulaciones obtenidos en la investigación de cada una de las partes que justifican la evolución del algoritmo planteado, y que permiten obtener conclusiones que se aplican al mismo. Sin embargo el trabajo de fusión no ha hecho más que empezar con este desarrollo, pudiéndose incluso mejorar en algún caso los resultados obtenidos con modificaciones del algoritmo y nuevos planteamientos de sistema modelado.

2. Algoritmos de fusión en sistemas de posicionamiento de robots móviles

En aplicaciones de navegación autónoma de robots móviles, una de las informaciones más importantes que maneja el controlador del robot es el conocimiento de la posición en que éste se encuentra en cada momento respecto a un sistema de referencia. De la precisión con que se conozca esta medida depende la funcionalidad del robot y el éxito de su tarea de navegación.

La estimación más sencilla de la posición del vehículo se lleva siempre a cabo mediante sensores internos (odométricos o inerciales) que proporcionan información relativa de la posición del robot a intervalos regulares de tiempo respecto a un punto de origen. Sin embargo su carácter relativo hace que la estimación de posición realizada empeore a medida que aumenta la distancia recorrida por el robot, debido tanto a inexactitudes de su modelo dead-reckoning, como a características impredecibles del terreno que recorre (se presenta un estudio más detallado sobre este tema en apartados siguientes) [Borenstein&Feng-96]. El empleo de sensores externos absolutos permite **corregir** la posición, logrando así una estimación más exacta de la posición del robot. Sin embargo, tal y como se ha comentado, esta información es complementaria a la de dead-reckoning, que se emplea en cualquier caso debido al mayor tiempo de procesamiento que requieren los sensores absolutos para obtener información de posición del robot.

Así ocurre en todos los trabajos estudiados en este campo: [PozoRuz-01], [Bonnifait&García-98], [Fabrizi&al.-00], entre otros. En todos ellos se usa el modelo inercial del móvil como descriptor de la generación de movimiento (evolución del vector de estados del sistema), y esta información se fusiona con la de posición absoluta proporcionada por sensores de visión, ultrasonidos o GPS en intervalos mayores de tiempo.

Con todo ello, en este punto se realiza un breve recorrido por las diversas técnicas de fusión, prestando mayor detalle a los métodos probabilísticos, y más específicamente al método del KF, que es el utilizado en el desarrollo de este trabajo.

2.1. Técnicas de integración sensorial

Los algoritmos matemáticos empleados para realizar la combinación de la información sensorial para el posicionamiento de robots móviles son muy diversos. Los más importantes, o al menos los más recurrentemente usados por los distintos grupos de investigación estudiados en esta investigación se presentan a continuación de forma resumida[2].

a) Fusión de datos mediante algoritmos de comportamientos o lógica borrosa:

Una de las aplicaciones más típicas de control borroso es, justamente, la fusión de datos. Existen diferentes técnicas específicas dentro de las basadas en comportamientos, sobre todo dentro del área de la inteligencia artifcial, y numerosos trabajos de investigación han sido realizados con éxito en este sentido, con autores como Arkin o Boem, también en el campo del posicionamiento para robots móviles.

Una de las técnicas, con aplicación de posicionamiento, más interesantes es la conocida por el término inglés de "Voting Fusion". Se trata de un método que no consiste en combinar medidas sino decisiones, y en el que no es necesario un modelo matemático concreto del sistema sensorial. En este tipo de técnicas de fusión existe un procesador que obtiene una respuesta de posición independiente para cada sensor, caracterizada por una función de probabilidad que se obtiene analizando las medidas proporcionadas por el sensor y su contenido de ruido.

Estas salidas de "posición probable" entran entonces en un esquema de selección que puede ser muy variado (selección del más votado, combinación de todas ellas, etc.) si bien suelen basarse en técnicas booleanas que finalmente terminan resultando una combinación borrosa.

Diferentes trabajos han puesto en práctica estos métodos en aplicaciones genéricas de clasificación de formas en el entorno (para aplicaciones de mapeo) y en guiado de robots móviles, fundamentalmente en tareas de salvar obstáculos. En [Pirjanian-98] se implementa un robot móvil que fusiona mediante esta técnica información de dos cámaras fijas (una orientada hacia el campo de visión izquierdo de avance del robot, y otra hacia el derecho, VISIÓN-L y VISIÓN-R respectivamente) y un SONAR. En la Figura 2.1 se muestra un ejemplo de actuación del móvil en una tarea de seguir la pared evitando obstáculos, así como los resultados cuantificados del experimento.

Las ventajas más importantes de este método radican en el hecho de no emplear un modelo matemático generalmente no lineal y estocástico complejo, por lo que el procesamiento es más rápido. Además como permite una organización paralela de la arquitectura de procesamiento la velocidad es aún mayor, y se facilita la incorporación de nuevos sensores en cualquier instante.

Sin embargo, por otro lado, el hecho de no emplear modelos conlleva una importante pérdida de precisión, tal y como se aprecia en la tabla de la anterior Figura 2.1, alcanzándose tasas de fiabilidad normalmente inferiores al 90%.

[2] La información ha sido obtenida del Seminario de Fusión de Datos desarrollado por el "Perceptual Computing and Comupuer Vision Group", del Instituto Federal de Tecnología de Zurich, en diciembre del año 2001. www.vision.ethz.ch/datafusion_seminar/dfseminar.html

SENSOR	Nº aciertos	Nº fallos	% fiabilidad
VISIÓN-L	14	11	56%
VISIÓN-R	15	10	60%
SONAR	12	13	48%
Voting	20	5	80%

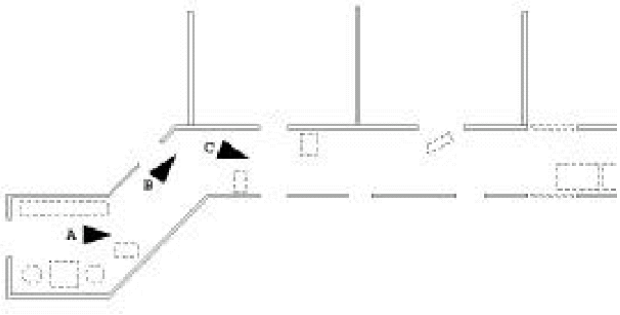


Figura 2.1. Técnica “Voting Fusion”. Experimento de [Pirjanian-98]

Las técnicas basadas en redes neuronales tienen un fundamento semejante, pero suelen tener otro tipo de aplicación menos orientada al posicionamiento de robots móviles. Autores como Zarzala o Laerhover han desarrollado trabajos relacionados con este tipo de técnicas de fusión, que pueden ser consultados para obtener más información sobre estos métodos.

b) Fusión probabilística

Son métodos que tienen en cuenta el modelo sensorial del sistema a posicionar y en los que para realizar la fusión se debe tener en cuenta la naturaleza y la magnitud de los errores que afectan a las medidas realizadas por los sensores. Es decir, no sólo es necesario un modelo del comportamiento cinemático del robot, sino también uno del error asociado a las medidas de los sensores, generalmente mediante una matriz de covarianza.

Dentro de este tipo de técnicas se pueden a su vez distinguir varias, en función de cual sea el algoritmo de fusión y de tratamiento del ruido utilizado. Sin embargo, quizás unas de las más interesantes sen las técnicas robustas de fusión [McKendall-91]. Estos algoritmos se basan en la minimización de la probabilidad de que aparezca un error intolerable en la obtención de la posición. Mediante el operador “minimax” se minimiza la máxima probabilidad de error. El problema de estas técnicas es que resultan computacionalmente muy costosas para ejecutar en tiempo real.

c) Modelos de mínimos cuadrados

Se va a hacer especial hincapié en este tipo de técnicas, pues son las más indicadas para la aplicación concreta que se persigue en este trabajo de investigación. Por ello a continuación se realiza una exposición general sobre el fundamento y características más importantes sobre las mismas, así como los diferentes algoritmo de fusión que se derivan del mismo planteamiento.

Estas técnicas recursiva también parte de la condición de que se conoce el modelo sensorial del movimiento del robot (obtenido generalmente siempre a partir de dead-reckoning), que este es lineal, y que las medidas de los diferentes sensores están afectadas por ruidos blancos (autocorrelación nula), de media cero, gaussianos (función de densidad de probabilidad -PDF- en forma de campana invertida) e incorrelados entre sí.

La segunda premisa simplifica en gran medida el proceso de fusión (si bien también se puede plantear con ruido coloreado [PozoRuz-01]), tanto es así que en caso deque el ruido que afecte a una de las medidas a combinar no sea blanco, se divide en parte blanca y parte coloreada, y la coloreada se incluye en el modelo para hacer que el ruido a tener en cuenta en el modelo sea blanco [Chui-90].

Con respecto a las premisas de ruidos de media nula e incorrelados, en la mayor parte de las veces las cumplen los sistemas sensoriales más habituales en robótica móvil. Además, en caso de no ser así, el problema se puede resolver sin mayor dificultad: mediante una calibración adecuada se elimina el valor medio del ruido, y mediante un adecuado desacoplamiento de los sensores.

El fundamento y la simplificación de las premisas se muestran en el capítulo siguiente, concretando la explicación para las técnicas de fusión basadas en el Filtro de Kalman, que son en definitiva las de interés en este trabajo.

Además, con respecto a la premisa de linealidad del modelo del sistema sensorial, que en la mayor parte de los casos no va a ser posible cumplir, se solventa extrapolando el algoritmo de fusión concreto al caso de que se linealice el modelo, generalmente mediante una serie de Taylor simple. Esto es lo que ocurre con el Filtro de Kalman Extendido, que es una evolución del Filtro de Kalman obtenida mediante una linealización como la comentada.

El mayor inconveniente de estos algoritmos de fusión, tal y como se comentará más adelante, se debe al hecho de que sea necesario un modelo más o menos exacto tanto del sistema sensorial, como del ruido que afecta a los datos sensados. En cualquier caso, son los modelos más veces recurridos en las distintas tareas de fusión implicadas en la robótica móvil, pudiéndose distinguir fundamentalmente dos técnicas básicas:

- **Técnicas básicas de mínimos cuadrados** [EasonGonzalez-91], que a partir del modelo lineal del sistema y de la suposición de ruidos blancos (y de media nula), gaussianos (normales) y no correlados entre sí, que afectan a las medidas de los distintos sistemas sensoriales, permite obtener la matriz de estimación que minimiza el error de observación (por ello toma el nombre de algoritmo de estimación óptima).

  Este algoritmo incluye variantes que contemplan, entre otros aspectos, el hecho de que el ruido que afecta a la salida no sea blanco (aplicándose en ese caso el teorema de Gauss-Markov), o que el sistema sea no lineal (realizando entonces una aproximación mediante la serie de Taylor). Además también se aplica la técnica de minimizar el valor medio del error cuadrático, si bien este método suele obtener peores resultados.

- **Técnicas de Filtro de Kalman (KF)** [Whelch&Bishop-01], también son un caso particular de técnicas de mínimos cuadrados, sólo que en este caso se trata de aplicar el algoritmo de probabilidad condicional (de ahí que muchas veces se le mencione como algoritmo Bayesiano) a las diferentes medidas y modelos sensoriales, conociendo también a priori el tipo de ruido y su matriz de covarianza. El KF podría definirse como un estimador óptimo (como el anterior) recursivo para sistemas lineales con ruido asociado a las medidas de características ya comentadas.

La ventaja del KF respecto a la técnica básica de mínimos cuadrados radica fundamentalmente en el hecho de que al tratarse de un algoritmo recursivo no precisa de todos los datos recopilados a lo largo de la historia del proceso para obtener el resultado de fusión sensorial en cada instante.

En realidad, el algoritmo del KF es el más usado en la estimación de procesos estocásticos para tareas de posicionamiento de robots móviles. Esta técnica fue desarrollada por Rudolph E. Kalman en 1960, como algoritmo lineal y recursivo de filtrado (o estimación) digital de datos discretos [Maybeck-79]. Sin embargo no empezó a usarse hasta la década de los 90, inicialmente sólo para filtrado de ruidos, y posteriormente como algoritmo de fusión en aplicaciones de navegación y de tracking.

3. El Filtro de Kalman Extendido (EKF) como algoritmo de fusión

Como se ha comentado en el capítulo anterior, el Filtro de Kalman Extendido es un estimador óptimo recursivo que permite fusionar medidas de posición de dos sistemas sensoriales, conocido el modelo no lineal de la planta a partir de las medidas de uno de ellos y la magnitud de la covarianza del error que afecta a todas las medidas.

En este punto se presenta el desarrollo del Filtro de Kalman para sistemas lineales a partir de la expresión de la probabilidad condicional, y se extrapola a sistemas no lineales, como el que interesa en este trabajo, para implementar el EKF. Se observa cómo se aplica la algoritmia obtenida a la fusión de datos para el posicionamiento de robots móviles, y se plantean modalidades alternativas del EKF.

3.1. Introducción y conceptos sobre estimación probabilística

Gran parte de la teoría de control (así como los métodos probabilísticos y de mínimos cuadrados de fusión, tal y como se explicó en el capítulo anterior) se soporta en la existencia de modelos matemáticos que representan el comportamiento de los sistemas a controlar o controladores. Sin embargo ningún modelo matemático es exacto (solamente representa los aspectos dominantes o críticos de la respuesta del sistema) debido a varias causas:

- Hay ciertos efectos que no son fácilmente modelables sino sólo aproximados matemáticamente,
- los sistemas se ven afectados por diferentes ruidos o modificaciones de sus parámetros que no se tienen en cuenta en el modelo,
- los sensores no proporcionan tampoco una información exacta sobre el estado de un sistema, por lo que las salidas no suelen corresponderse con la modelada,

Estas razones obligan en muchos casos a incorporar estimadores que permitan, no ya conocer ciertas variables del sistema a controlar que de otro modo no serían conocidas, sino también obtener una medida más fiable de variables que aunque se puedan sensar, están muy contaminadas por las distintas fuentes de error.

Como ya se ha comentado en la introducción, el algoritmo de Kalman es un filtro digital que minimiza el error de estimación de forma recursiva gracias a la aplicación de técnicas probabilísticas sobre el modelo del sistema sensorial a filtrar y el ruido asociado a dicho modelo.

Los métodos probabilísticos de estimación (como el KF) se basan también en el conocimiento a priori del comportamiento del sistema es decir, en modelos como los comentados. Esto provoca que las técnicas de estimación basadas en modelos tengan su fiabilidad socavada por los errores de modelado. Sin embargo, la técnica del KF, incorpora la ventaja de estimar la evolución del sistema, compararla con la salida real y mejorar su estimación según va ejecutándose (por eso es recursivo).

Para mejorar el funcionamiento de la estimación, este algoritmo incorpora la siguiente información acerca del sistema a estudiar:

- Conocimiento del modelo matemático discreto del sistema a estudiar. Se supone ya afectado por cierto ruido de medidas (v_k) y de modelado o de estados (w_k).

$$\begin{aligned} x_k &= f\left(x_{k-1}, u_k, w_k\right) \\ z_k &= h\left(x_k, v_k\right) \end{aligned} \qquad \text{<3.1>}$$

- Descripción estadística del ruido o error aplicado a las medidas del estado (o debido a la inexactitud del modelo planteado) y de la salida del sistema, respectivamente.

$$\begin{aligned} p(w) &\approx N\left(\mu_w, Q\right) \\ p(v) &\approx N\left(\mu_v, R\right) \end{aligned} \qquad \text{<3.2>}$$

Con esta información que describe completamente el sistema (si se cumplen la serie de condiciones ya presentadas sobre las características de los dos elementos modelados), se realiza la estimación mediante un proceso de **predicción y posterior corrección** que pretende minimizar la covarianza del error de estimación. De este modo se consigue la característica de **estimación óptima** que incluye el KF.

Además el algoritmo es **recursivo**, tal y como ya se ha comentado, por lo que su utilidad en robótica se centra fundamentalmente en tareas de fusión de información de la posición (o tareas de tracking) y no tanto, por ejemplo, en tareas de fusión de mapas, donde se suelen emplear otras técnicas probabilísticas [Marksacov-95] que requieren de un análisis continuo de todo el banco de datos sensados.

Con respecto a las condiciones que se han de cumplir para poder aplicar este algoritmo de estimación óptima (modelo lineal y ruidos gaussianos, blancos, de media nula e incorrelados entre sí), ya se ha comentado que aunque a simple vista parecen limitar demasiado el rango de aplicación del algoritmo, son sin embargo sencillas de cumplir:

- Si bien la condición de linealidad no suele cumplirse en la práctica totalidad de los sistemas (de hecho así ocurre con el sistema de interés), es posible realizar una extensión del estimador, a base de linealizar el modelo mediante un desarrollo de Taylor alrededor del punto de trabajo, que no será otro que el punto de estimación obtenido por el filtro. Este tratamiento dará lugar a la extensión del algoritmo de filtrado, el EKF que será además el empleado en la aplicación de interés.

- La mayor parte de los procesos naturales continuos tienen un comportamiento normal (de ahí el nombre de la PDF). Esta cuestión puede ser justificada físicamente por el hecho de que normalmente los ruidos a las medidas se deben a diversas pequeñas fuentes de ruido. Matemáticamente, si un conjunto de variables aleatorias independientes se suman el resultado viene descrito mediante una PDF gaussiana, independientemente de la forma de las PDFs independientes (“teorema del límite central”).

- La eliminación del valor medio del ruido que afecta a todas las medidas ($\mu_w = 0$ y $\mu_v = 0$) se realiza mediante técnicas de calibración sencillas, y es fundamental para simplificar el funcionamiento del estimador, como se verá más adelante. En el sistema de interés en este trabajo los métodos de calibración serán expuestos en el capítulo siguiente del documento y consisten en tratamientos off-line del sistema sensorial.

- Si bien la condición de autocorrelación es más difícil de cumplir, también suele cumplirse al menos en el ancho de banda de trabajo del modelo utilizado. Además en caso de que esta condición no se cumpla es posible separar la parte coloreada del ruido e incorporarlo al modelo del sistema [Chui-90], o realizar un estudio más complejo del desarrollo del algoritmo [PozoRuz-01], tal y como ya se ha explicado. Finalmente, también aparece una premisa de anulación de la correlación cruzada entre los ruidos de medida y de estado, que como ya se ha comentado suele cumplirse sin ningún problema en aplicaciones del tipo de la que aquí se estudia.

Realizando estas suposiciones básicas (que serán analizadas específica para la aplicación de interés en el capítulo de modelado) el algoritmo del KF se simplifica en gran medida, tal y como se desarrolla en el siguiente apartado. Además, es interesante comentar en este punto, que aunque en la mayor parte de las aplicaciones estudiadas en el desarrollo de este trabajo, las condiciones establecidas raramente se cumplen en su totalidad, el algoritmo suele dar buenos resultados de estimación independiente de este hecho (se demostrará en el último capítulo del documento).

3.2. Desarrollo del algoritmo recursivo de estimación óptima

Tal y como ya se ha explicado, la base del KF es la aplicación de la ley de la probabilidad condicional (regla de Bayes) sobre el modelo del sistema que se presentó en las ecuaciones <3.1> y <3.2>. Simplificando la ecuación del modelo del sistema (lineal y observable), y suponiendo su comportamiento como el de una simple variable aleatoria, se obtiene:

$$\begin{aligned} \dot{x}(t) &= A \cdot x(t) + B \cdot u(t) + w(t) \\ z(t) &= C \cdot x(t) + v(t) = x(t) + v(t) \end{aligned} \quad \text{<3.3>}$$

El desarrollo del KF podría partir del hecho de que en un instante t_1 se predice, a través de su modelo, el comportamiento de un sistema (variable aleatoria), con un resultado de estado (o de salida, tal y como se presenta en la ecuación <3.3>) caracterizado por una PDF de media x_1 y varianza σ_{w1}^2 o simplemente σ_1^2. En el instante t_2 se mide dicho comportamiento mediante un sensor que proporciona solamente información de salida, con un resultado de media z_2 y varianza σ_v^2 (incorrelado con el anterior). La probabilidad condicional de que el comportamiento sea el estimado (x_1) midiendo z_2 se obtiene mediante la mencionada regla de Bayes, obteniéndose como resultado una PDF gaussiana de media x_2 y varianza σ_2^2 (ver figura 3.1):

$$\mu = x_2 = \left[\frac{\sigma_v^2}{\left(\sigma_1^2 + \sigma_v^2\right)}\right] x_1 + \left[\frac{\sigma_w^2}{\left(\sigma_1^2 + \sigma_v^2\right)}\right] z_2 \qquad <3.4>$$

$$\frac{1}{\sigma_2^2} = \frac{1}{\sigma_v^2} + \frac{1}{\sigma_1^2}$$

La mejor estimación de dicho comportamiento es, por tanto, μ (los ruidos son normales y de media nula) que se podrá poner como la estimación óptima a posteriori, y que agrupando términos supondrá:

$$x_2 = x_1 + \left[\frac{\sigma_1^2}{\left(\sigma_1^2 + \sigma_v^2\right)}\right](z_2 - x_1) \quad \Rightarrow$$

$$x_2 = x_1 + K_2(z_2 - x_1), \quad K_2 = \left[\frac{\sigma_1^2}{\left(\sigma_1^2 + \sigma_v^2\right)}\right] \qquad <3.5>$$

De esta ecuación se obtiene por tanto la matriz de Kalman (K_2) a partir de la varianza del error del modelo o del error de estimación σ_1^2 y de la varianza del ruido de medidas σ_v^2. Y la mejor estimación se obtendrá corrigiendo la obtenida del modelo del sistema (x_1) mediante un término (z_2 - x_1), que se denomina residuo y que está ponderado por la matriz de estimación óptima K_2.

Este sencillo ejemplo, permite observar cómo la incorporación de la probabilidad condicional permite que la incertidumbre de estimación se vaya reduciendo ($\sigma_2^2 < \min\left[\sigma_v^2, \sigma_1^2\right]$), según la ecuación <3.4>, al incorporar recursivamente más información al proceso de estimación. Además, se puede obtener otra conclusión de este análisis, y es que por muy mala que sea la medida (σ_v^2) aporta una mejora en cualquier caso a la estimación final de la variable de interés (ver figura 3.1).

También se puede observar en este planteamiento básico, el modelo **predictor-corrector** del algoritmo: la salida estimada será inicialmente la predicha por el modelo (x_1), que será corregida (x_2) por el factor de residuo (z_2-x_1) y ponderada por la ganancia de Kalman (K_2) en el instante siguiente (t_2).

Un último análisis se va a presentar con respecto a este ejemplo base. Rescribiendo la ecuación de la varianza presentada en la ecuación <3.4>, mediante el uso de la matriz de estimación (K_2) se puede poner:

$$\sigma_2^2 = \sigma_1^2 - K_2\sigma_1^2 \qquad <3.6>$$

Lo cual da una idea de cómo la matriz de varianza de error de predicción se va actualizando también gracias a la matriz de Kalman.

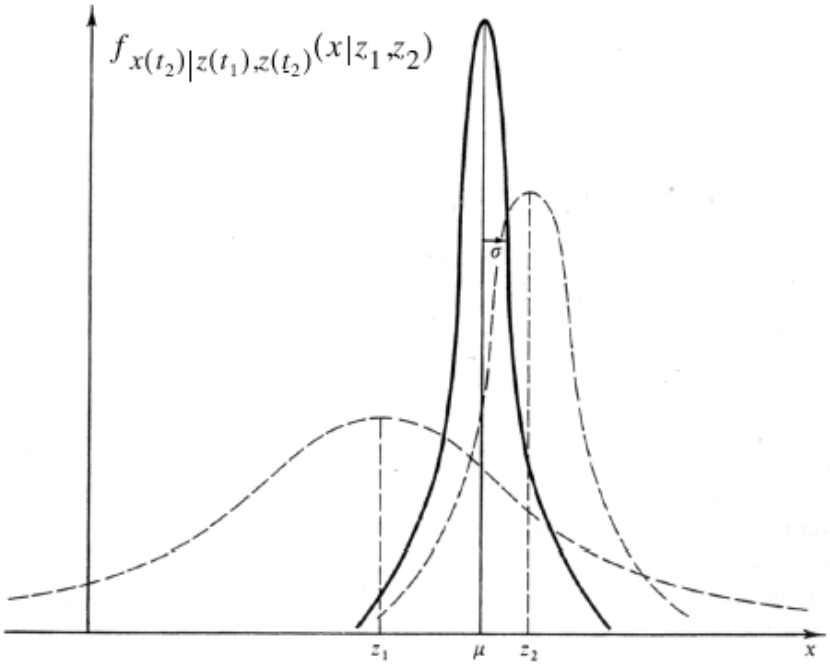


Figura 3.1. Representación gráfica del funcionamiento del KF como estimador basado en el modelo del ruido que afecta a las medidas

El algoritmo puede ser expresado matemáticamente de forma tan sencilla gracias a las condiciones impuestas a las fuentes de ruido que se incluyen en el vector de estados y de medida. En caso de que las fuentes de ruido no fuesen gaussianas o blancas, de media nula e incorreladas entre sí, el algoritmo se complicaría por lo que diferentes técnicas han sido planteadas para atajar estos problemas. Al final de este capítulo se presentan algunas de las soluciones propuestas para situaciones en las que no se den las condiciones impuestas a las fuentes de error por el planteamiento Bayesiano.

3.3. El algoritmo del KF para sistemas discretos

En el apartado anterior se ha presentado, mediante un ejemplo sencillo, cual es la base de funcionamiento del estimador. En este apartado se presenta el desarrollo del mismo para un sistema estocástico en lugar de para una variable aleatoria. En este caso se definen matrices de covarianza en lugar de varianzas (ruido blanco y gaussiano), y para hacer el caso más general se parte del modelo lineal completo del sistema discreto:

$$\begin{aligned} x_k &= A \cdot x_{k-1} + B \cdot u_k + w_k \\ z_k &= C \cdot x_k + v_k \end{aligned} \quad , \quad \begin{aligned} p(w) &\approx N(0,Q) \\ p(v) &\approx N(0,R) \end{aligned} \qquad <3.7>$$

Aunque en el caso general se establece que las matrices A, B, C, Q y R son constantes (sistemas invariantes y ruidos estacionarios) la extrapolación del algoritmo a matrices variables no sería muy complejo, por lo que se plantea el caso más simple.

Además, para comprender la nomenclatura referida a las dos etapas de evolución del algoritmo (predicción y corrección) se definirá $m_{k+1/k}$ como la estimación de la variable m en el instante k+1 a partir de la información existente en el instante k (predicción) y $m_{k+1/k+1}$ como la estimación de la variable m en el instante k+1 a partir de la información existente en el instante k+1 (corrección).

El primer paso en la evolución del algoritmo es el de definir la matriz de covarianza del error de estimación P (que es la que ha de minimizarse para conseguir un estimador óptimo) como:

$$\begin{aligned} P_{k+1/k} &= E\left[e_{k+1/k} \cdot e_{k+1/k}\right] \\ P_{k+1/k+1} &= E\left[e_{k+1/k+1} \cdot e_{k+1/k+1}\right] \end{aligned} \quad , \quad \begin{aligned} e_{k+1/k} &= x_{k+1} - \hat{x}_{k+1/k} \\ e_{k+1/k+1} &= x_{k+1} - \hat{x}_{k+1/k+1} \end{aligned} \qquad <3.8>$$

Merece la pena destacar la diferencia entre x_k y $\hat{x}_k$. La primera se refiere al valor del vector de estado obtenido mediante el modelo matemático del sistema, y la segunda a la predicción que se realiza del vector de estado con el KF. En la aplicación del KF a tareas de fusión el significado cambia ligeramente tal y como se verá en capítulos siguientes.

Con todo ello, y siguiendo el razonamiento Bayesiano antes presentado (ver ecuación <3.5>), la estimación a posteriori del estado se obtendrá del siguiente modo:

$$\hat{x}_{k+1/k+1} = \hat{x}_{k+1/k} + K_{k+1}(z_{k+1} - C \cdot x_{k+1/k}) \qquad <3.9>$$

La ganancia de Kalman se calcula minimizando la matriz de covarianza del error de estimación $P_{k+1/k+1}$, obteniéndose el siguiente valor:

$$K_{k+1} = \frac{P_{k+1/k} \cdot C^T}{C \cdot P_{k+1/k} \cdot C^T + R} \qquad <3.10>$$

Se observa claramente que la ecuación <3.10> tiene un aspecto semejante al presentado en la anterior <3.5>, sólo que se ha cambiado la varianza por la matriz de covarianza y se ha añadido la matriz C (antes supuesta 1).

Finalmente, la matriz de covarianza del error de estimación de estados definida ya anteriormente, se obtiene a partir de las expresiones anteriores (que también recuerda a la anterior ecuación <3.6>):

$$P_{k+1/k+1} = P_{k+1/k} - K_{k+1} \cdot C \cdot P_{k+1/k} \qquad <3.11>$$

Con todo esto se completa el algoritmo buscado, que en la siguiente figura se organiza por ecuaciones en las dos etapas básicas en que ha de desarrollarse (predicción y corrección):

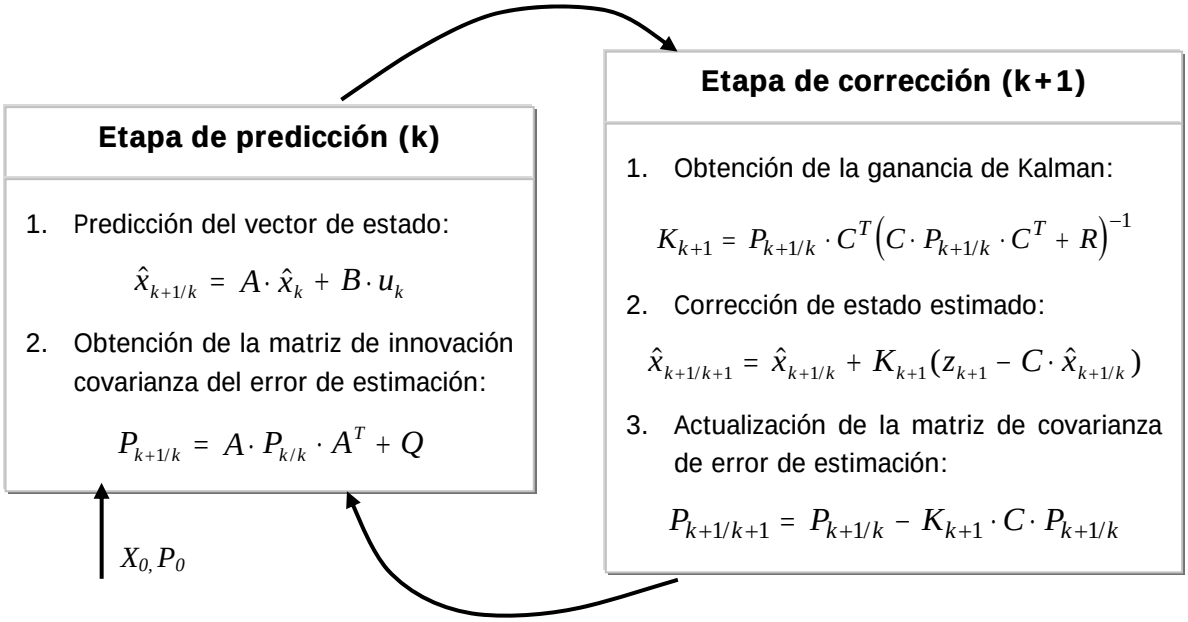


Figura 3.2. Diagrama explicativo del desarrollo del Filtro de Kalman en dos etapas (KF)

En la etapa de predicción (en el instante k) se obtiene el valor del vector de estados estimados ($x_{k+1/k}$) empleando para ello el modelo matemático del sistema, así se obtendrá la predicción del vector de estados. Además se obtiene el valor de innovación de la matriz de covarianza del error del predicción ($P_{k+1/k}$), que incluye la matriz de covarianza de error del sistema (Q), incorporándose así, a la matriz P, el error de sensado (o de modelado).

La etapa de corrección (en el instante k+1) comienza con la obtención de la ganancia de Kalman a partir de las matrices de covarianza de error de estimación innovada ($P_{k+1/k}$) y de covarianza del error de medidas (R). Con ella se corrige la estimación del estado ($\hat{x}_{k+1/k+1}$) del sistema y finalmente se actualiza la matriz de covarianza del error de estimación ($P_{k+1/k+1}$). Merece la pena destacar que este paso consiste fundamentalmente en fusionar la información obtenida por el modelo del sistema con información externa del sensado de la salida. Este hecho es el que dará pie a emplear el algoritmo en procesos de fusión.

Con estos datos se volvería a realizar el proceso de predicción, ahora en el instante k+1 para obtener la estimación del instante k+2, y así consecutivamente.

Como ya se ha comentado, la recursividad del algoritmo representada de este modo evita tener que realizar cálculos de estimación con toda la batería de datos de comportamiento de instantes anteriores, tal y como ocurre en otras técnicas de filtrado (o de fusión asociadas). De este modo se agiliza en gran medida el algoritmo de estimación o filtrado, haciéndolo válido para aplicaciones de navegación o posicionamiento de

robots móviles como el que aquí se presenta (en las que otras técnicas probabilísticas no tienen aplicación, debido a su lentitud de proceso).

Con esto quedaría totalmente detallado el funcionamiento del estimador óptimo. Merece la pena hacer especial hincapié en la inicialización del algoritmo, que, tal y como se observa en el diagrama de la figura 3.2, pasa por la inicialización de las matrices de estado (x_0) y de covarianza del error de predicción (P_0). Si bien la primera suele ser fácil de fijar (en los casos de navegación de robots esta matriz está relacionada con la posición inicial del mismo, que en principio se supone conocida) la segunda no lo es tanto. Hay varias tácticas para asignarle un valor inicial.

- Si se conoce x_0, P_0 será nulo, ya que representa la incertidumbre del valor de estimación inicial.

- Si no se conoce x_0, P_0 deberá fijarse en función de su significado, pero en cualquier caso una técnica muy recurrida es la de ejecutar el estimador off-line (mediante una simulación) y ajustarlo así por prueba y error.

- Existen métodos matemáticos que permiten obtener el valor final del P_0, como el de [Grewal&Andrews-01], sin embargo solamente es válido para casos en los que R y Q son constantes, en los que el KF converge fácilmente independientemente del valor inicial de P.

En cualquier caso, si el sistema es lineal e invariante el estimador recursivo suele converger siempre (más o menos rápidamente) independientemente del valor de P_0. Será por tanto más crítica la elección de este parámetro a la hora de implementar el estimador para sistemas no lineales.

En lo que a las matrices Q y R se refiere, al ser la covarianza del error de sistema y de medidas se pueden obtener fácilmente, al menos de forma aparente ($Q = E\left[w_k \cdot w_k^T\right], R = E\left[v_k \cdot v_k^T\right]$, aunque v_k y w_k aparecen dependientes del tiempo Q y R suelen considerarse constantes). Sin embargo, si bien la matriz R puede obtenerse de forma empírica realizando medidas off-line de la salida y obteniendo dicho parámetro estadístico, no ocurre de igual modo con la matriz Q, ya que no suele ser habitual tener acceso a las variables de estado del sistema (si bien esto sí ocurre en las aplicaciones de fusión). En caso de no poder obtener Q de forma matemática se fijará por tanteo (generalmente también mediante sucesivas pruebas del estimador realizadas off-line). En ese caso es importante tener en cuenta ciertos aspectos sobre el valor de estas matrices.

- La relación de magnitud entre Q y R es indicativa de la fiabilidad de los dos elementos de medida o modelado. Así, si Q<R significa que el modelo proporciona más fiabilidad sobre el vector de estado que la matriz de salida. En ese caso la matriz P tomará valores pequeños, así como la ganancia de Kalman. Y si R<Q significará que el modelo (o el sistema de sensado del vector de estado) es menos fiable que los sensores de salida (P y K serán pequeñas en la evolución del algoritmo).

- Ambas matrices son cuadradas y diagonales (pues el ruido no ha de estar autocorrelado, ha de ser blanco). En caso de que los experimentos de obtención de las matrices devuelvan algún valor no nulo fuera de la diagonal principal el ruido será coloreado, y será necesario emplear alguna técnica para corregir este aspecto. Si alguna de las fuentes de ruido presenta fuentes sistemáticas, la media de estas

variables aleatorias no será nula, por lo que será necesario eliminarlos de algún modo. Ambos aspectos se verán de forma más detallada en el capítulo siguiente al especificarlos para las fuentes sensoriales de interés en este trabajo.

- Finalmente las dos fuentes de ruido (v_k y w_k) ha de estar incorreladas entre sí, lo cual solamente suele no cumplirse en aplicaciones de fusión en las que los elementos de sensado son conjuntos o están relacionados de algún modo. Si esto es así será necesario desacoplarlos previamente para poder implementar el KF.

3.4. El algoritmo del EKF para sistemas discretos y no lineales

El EKF surge de la necesidad de aplicar el estimador óptimo de Kalman a sistemas regidos por un modelo no lineal. La idea básica de la implementación de esta nueva versión del estimador consiste en linealizar el algoritmo alrededor del punto de trabajo mediante una serie de Taylor.

El análisis que permite desarrollar este nuevo algoritmo sigue los mismos pasos que el presentado en el punto anterior, sólo que el modelo de partida del sistema sensorial a observar cambia (en este caso no es lineal), por lo que será necesario realizar ciertas transformaciones previas para seguir el mismo planteamiento.

En este caso, la representación general del modelo del sistema queda, tal y como se mostraba al principio del documento (ecuación <3.1>):

$$\begin{aligned} x_k &= f\left(x_{k-1}, u_k, w_k\right) \\ z_k &= h\left(x_k, v_k\right) \end{aligned} \qquad <3.12>$$

Para linealizar el modelo primeramente hay tener en cuenta que del modelo anterior no se conoce el valor instantáneo del error por lo que no se puede incluir en el modelo de estimación y ha de rescribirse del siguiente modo:

$$\begin{aligned} \tilde{x}_k &= f\left(\hat{x}_{k-1}, u_k, 0\right) \\ \tilde{z}_k &= h\left(\tilde{x}_k, 0\right) \end{aligned} \qquad <3.13>$$

Linealizando el sistema de ecuaciones anterior, tal y como se ha dicho, queda:

$$\begin{aligned} x_k &= \tilde{x}_k + A_k \cdot \left(x_{k-1} - \hat{x}_{k-1}\right) + W_k \cdot w_{k-1} \\ z_k &= \tilde{z}_k + C_k \cdot \left(x_k - \tilde{x}_k\right) + V_k \cdot v_k \end{aligned} \qquad <3.14>$$

Donde:

- x_k y z_k son los valores y medidas reales del vector de estado y de salida, respectivamente.
- $\tilde{x}_k$ y $\tilde{z}_k$ son los valores estimados sin error (procedentes del modelo) del vector de estado y de salida, respectivamente.

- $\hat{x}_k$ es el valor del vector de estado estimado y corregido a posteriori ($\hat{x}_{k+1/k}$).
- A_k es el jacobiano de derivadas parciales de $f()$ con respecto a x_k en el instante k.

$$A_k \Rightarrow A_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}}\left(\hat{x}_{k-1}, u_k, 0\right) \quad \text{<3.15>}$$

- C_k es el jacobiano de derivadas parciales de $h()$ con respecto a x_k en el instante k.

$$C_k \Rightarrow C_{[i,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}}\left(\hat{x}_k, u_k, 0\right) \quad \text{<3.16>}$$

- W_k es el jacobiano de derivadas parciales de $f()$ con respecto a w_k en el instante k.

$$W_k \Rightarrow W_{[i,j]} = \frac{\partial f_{[i]}}{\partial w_{[j]}}\left(\hat{x}_{k-1}, u_k, 0\right) \quad \text{<3.17>}$$

- V_k es el jacobiano de derivadas parciales de $h()$ con respecto a v_k en el instante k.

$$V_k \Rightarrow V_{[i,j]} = \frac{\partial h_{[i]}}{\partial v_{[j]}}\left(\hat{x}_k, 0\right) \quad \text{<3.18>}$$

Merece la pena destacar el hecho de que en este caso las matrices A y C que eran constante en el KF básico no lo son en esta nueva formulación (será necesario recalcular su valor en cada paso de ejecución del estimador óptimo).

A partir de este planteamiento se define el error de predicción del vector de estados y de la salida de esta forma:

$$\begin{aligned} e_{x_k} &= x_k - \hat{x}_k \\ e_{z_k} &= z_k - \hat{z}_k \end{aligned} \quad \text{<3.19>}$$

Además, como no se tiene acceso directo al vector de estados real pero sí al de salida real, empleando las expresiones de la ecuación <3.14> se puede rescribir lo anterior como:

$$\begin{aligned} e_{x_k} &= A_k\left(x_{k-1} - \hat{x}_{k-1}\right) + \varepsilon_k \\ e_{z_k} &= C_k \cdot e_{x_k} + \eta_k \end{aligned} \quad \text{<3.20>}$$

Donde ε_k y η_k son dos nuevas variables aleatorias con media nula y varianza de valor $\sigma_{\varepsilon_k} = W_k Q W_k^T$ y $\sigma_{\eta_k} = V_k R V_k^T$, dado que v_k y w_k lo son.

La ecuación <3.20> anterior tiene un formato semejante al de descripción del modelo de un sistema lineal (ver ecuación <3.7>), por lo que se va a tomar como punto de partida para desarrollar el EKF. En este caso el vector de "estado" es el error de estimación del sistema e_{x_k}, que además se desea hacer nulo. Con todo ello la ecuación de regulación de este segundo estimador óptimo de Kalman será:

$$\hat{e}_{x_k} = K_k \cdot e_{z_k} \qquad <3.21>$$

Además, la ecuación de obtención de la estimación a posteriori del vector de estado original se define ahora como:

$$\hat{x}_k = \tilde{x}_k + \hat{e}_{x_k} \qquad <3.22>$$

Sustituyendo la ecuación <3.21> en ésta última, se obtiene ésta de una forma más familiar (semejante a la ecuación <3.9>):

$$\hat{x}_k = \tilde{x}_k + K_k \cdot e_{z_k} = \tilde{x}_k + K_k\left(z_k - \tilde{z}_k\right) \qquad <3.23>$$

Para completar el EKF, se definen la ganancia del estimador K_k de modo que minimice el error de predicción de e_{k_x} (el vector de estado en este nuevo modelo transformado), tal y como se presentó en la anterior ecuación <3.10>; y la matriz de covarianza del error de estimación P a partir de las anteriores, tal y como se presentaba en la anterior ecuación <3.11>. En este caso, en la expresión de las dos matrices comentadas se sustituye R y Q por sus equivalentes, obtenidas a partir de la expresión <3.20>.

Con todo ello, en la figura 3.3 se muestra como queda el algoritmo del estimador Bayesiano óptimo para sistemas no lineales, de forma equivalente a como se hizo con el KF básico en la anterior figura 3.2.

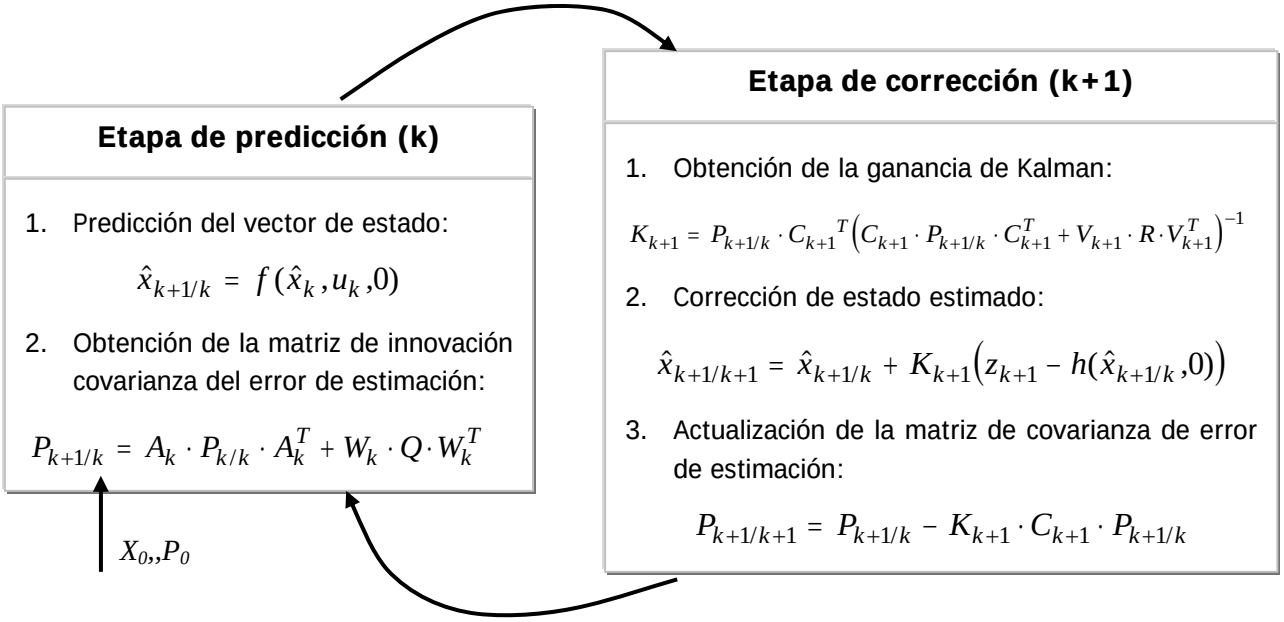


Figura 3.3. Diagrama explicativo del desarrollo del Filtro de Kalman Extendido (EKF)

Para mantener la presencia del concepto de estimación a priori y a posteriori, en las ecuaciones de la anterior figura se ha sustituido $\tilde{x}_k$ por $\hat{x}_{k+1/k}$. Además, se puede observar fácilmente en estas expresiones como A_k, C_k, W_k y V_k ahora sí que se han hecho dependientes del tiempo (debido a que su

origen es no lineal será necesario calcular su valor en cada instante mediante una serie de Taylor centrada en el punto de trabajo).

Es interesante observar que a la hora de linealizar el modelo del sistema a observar, se ha aplicado también la transformación a las fuentes de ruido que modificaban el vector de estados y de salida (w_k y v_k) obteniendo otras fuentes de ruido (ε_k y η_k). El problema es que esta transformación ha eliminado el carácter gaussiano de las fuentes primitivas, con lo que el planteamiento Bayesiano que da lugar a todo el desarrollo del estimador óptimo ya no es todo lo exacto que se desea. Este hecho produce que el EKF no converja con la misma facilidad que lo hace el KF, sino que lo hará dependiendo del valor inicial de la matriz de covarianza del error de estimación P o del modelo del sistema en general. Existen algoritmos derivados del EKF que tienen en cuenta este hecho, entre ellos el de [Julier&Uhlmann-96], que será comentado en el apartado siguiente.

3.5. Otras consideraciones y modalidades alternativas del EKF y su uso en tareas de fusión

Como ya se ha comentado el estimador presentado ha sido extensamente usado en el mundo de la robótica. Debido a ello, diversas mejoras del algoritmo para casos particulares han sido desarrolladas.

Un ejemplo de ello son las técnicas multimodales (o de múltiples modelos [BarShalom&Li-93]) que se emplean cuando no se posee un modelo exacto de la planta a estimar, sino varios aproximados, que a su vez pueden ser fijos o dinámicos. Estas técnicas se basan en desarrollar el KF para cada uno de los posibles modelos, y luego, en función de la covarianza de error de estimación de cada uno se combinan todos ellos para dar lugar a una estimación global (ver ecuaciones <3.24> y <3.25>).

$$\hat{x}_k = \sum_{j=1}^{r} p_j(k) \cdot \hat{x}_{k,\mu_j} \quad <3.24>$$

(μ representa a cada uno de los r modelos diferentes y $\hat{x}_k$ a la salida de estimación global)

$$\hat{P}_k = \sum_{j=1}^{r} p_j(k) \left[P_{k,\mu_j} + \varepsilon_{\mu_j} \cdot \varepsilon_{\mu_j}^T \right] \quad <3.25>$$

Donde $\varepsilon_{\mu_j} = \hat{x}_k - \hat{x}_{k,\mu_j}$, $p_j(k)$ es la probabilidad de que el modelo j sea el candidato adecuado para la estimación en el instante k (se crea una distribución gaussiana de probabilidad para cada modelo), y $\hat{P}_k$ es la matriz de covarianza del error de estimación global.

En el caso que interesa en este trabajo de investigación, esta alternativa no es necesaria pues el modelo inercial de la planta a estimar es bien conocido, pero sí lo sería para modelos de robots con una cinemática más compleja. Además, el método presentado es, en definitiva, también un método de fusión multisensorial, pero en el que se parte de varios modelos diferentes del sistema en cuestión (según el sensor de estados empleado).

En general las herramientas de estimación estocástica, como el KF, han sido desde su inicio de aplicación empleadas para fusionar información de distintos sensores. La idea de partida (concretamente en el estimador de Kalman) es utilizar la matriz de Kalman (ver ecuación <3.9> o <3.23>) para ponderar y asociar diferentes medidas del estado de un sistema, proporcionando así estimaciones más fiables y estables que si se emplease solamente un sensor. Esta aplicación específica del KF se vio representada por primera vez en lo que se dio a llamar “KF de realimentación indirecta”. En la figura 3.4 se muestra el principio de funcionamiento de este método, que coincide totalmente con los objetivos planteados en el trabajo que aquí se presenta.

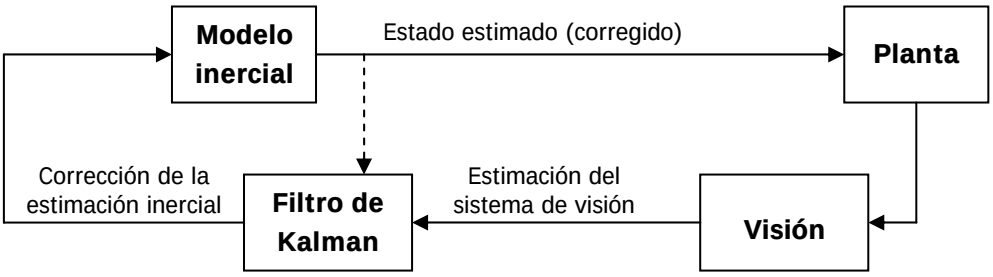


Figura 3.4. Representación del funcionamiento del KF en tareas de fusión. Arquitectura del llamado “KF de realimentación indirecta”

Merece la pena observar el hecho de que en esta configuración, el KF utiliza un único modelo de sistema, pero dos modelos de medidas, uno para el elemento inercial (que será además en este trabajo el que da origen al modelo de estados del sistema) y otro para el elemento redundante de medida (el elemento de visión en el caso que aquí se presenta).

El diagrama presentado en la figura 3.4 anterior da una idea más concreta sobre el funcionamiento del algoritmo de fusión que ha sido elegido en este trabajo. En él se aprecia claramente cuál es el papel de los dos sistemas sensoriales de posicionamiento que se desean combinar, y cómo el EKF puede lograr el efecto buscado.

Esta idea es la que ha de retenerse como resultado del desarrollo de todo el capítulo, teniendo en cuenta que la matemática será completada en el capítulo siguiente con el modelo específico de cada uno de los sistemas sensoriales implicados en la aplicación de interés.

Finalmente, para completar el estudio sobre el KF para la fusión de datos, no se puede dejar de comentar en este trabajo de investigación las aportaciones realizadas por [Julier&Uhlmann-96]. Se trata de volver a desarrollar el algoritmo del EKF, teniendo en consideración el hecho de que, al realizar la linealización del modelo del sistema y del ruido en el desarrollo del algoritmo, hay que plantear el hecho de que el ruido, considerado inicialmente gaussiano, deja de serlo. En ese caso se recalcula la matriz de covarianza asociada al ruido en cada instante, en lugar de realizar el cálculo de la matriz jacobiana que intenta adaptar el ruido caracterizado inicialmente a un nuevo sistema lineal. El resultado es el UKF, o Filtro de Kalman Descentralizado, que según los textos consultados en el trabajo de investigación arroja mejores resultados que el EKF a la hora de modelar ruidos gaussianos aplicados sobre sistemas no lineales. Se deja este punto como un camino abierto, por tanto, para posteriores trabajos de investigación.

4. Modelado del sistema de posicionamiento de la silla de ruedas automatizada

En este capítulo se presenta el modelo del sistema sobre el que se desea realizar la tarea de fusión. Una vez presentadas las bases de funcionamiento del EKF en el capítulo anterior, así como su aplicabilidad a tareas de fusión queda claro que han de ser modelados tanto la representación de los estados del sistema de posicionamiento, definidos por el sistema odométrico, como la salida en posición que se obtiene mediante el sistema de visión, así como los ruidos asociados a ambas medidas.

Al ser el sistema inercial el elegido para determinar la evolución de los estados, será a partir de estas medidas como se construirá el modelo cinemático del robot, mediante un proceso de dead-reckoning. El modelo obtenido es no lineal, lo cual da pie a consideraciones en su uso para la implementación del algoritmo de fusión, y obliga al uso de la versión extendida del KF (EKF). Además es necesario tener en cuenta que el procesado digital del estimador obliga a obtener modelos discretizados con un determinado periodo de muestreo.

Por su parte, en lo que al modelo de visión se refiere, en este documento se muestra cómo a partir de la información obtenida por el algoritmo SPL [García-01], se obtiene, mediante una sencilla transformación, la información de salida del modelo de posición del robot, que será incorporada durante el proceso de fusión a la estimada a partir de las medidas de odometría.

Con esta presentación y lo estudiado en el apartado anterior, ya se puede vislumbrar cómo se realizará la fusión de ambos sistemas de medidas mediante un EKF (ver figura 4.1). El elemento inercial, mediante el modelo de dead-reckoning no lineal desarrollado, permitirá obtener los estados estimados del sistema robótico (y a partir de estos las salidas de posición del mismo, que en este caso van a coincidir), mientras que el sistema de visión proporcionará mediante la transformación mencionada la salida que corregirá el estado estimado a través de la ganancia de Kalman, calculada para minimizar la covarianza de error de estimación.

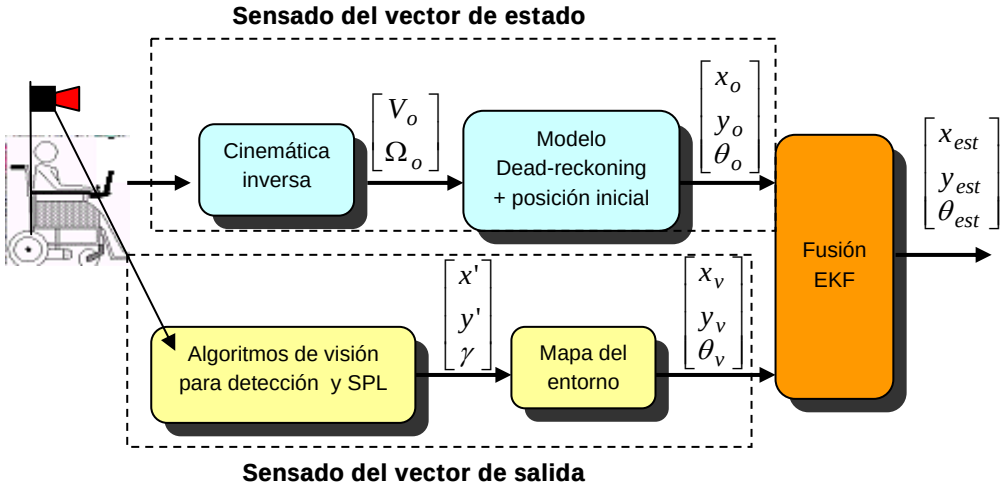


Figura 4.1[3]**.** Diagrama funcional del algoritmo de fusión para posicionamiento de un robot móvil mediante un EKF

Además, para poder desarrollar una estimación adecuada mediante un EKF ha sido necesario representar no sólo el comportamiento determinístico de los dos sistemas de posicionamiento sino también el aleatorio, que vendrá caracterizado por las matrices de covarianza de los ruidos asociados a cada una de las medidas o modelos.

Para abordar con facilidad el problema completo de modelado, se presentará en primera instancia el del sistema odométrico y posteriormente el del sistema de visión.

4.1. Modelo odométrico de la silla de ruedas

El modelo inercial del robot móvil objeto de la aplicación de fusión ha sido desarrollado en múltiples trabajos previos [Marrón-00]. La posición del mismo se obtiene mediante un proceso de integración de las medidas de odometría (ver figura 4.2) tal y como se reproduce a continuación:

$$\begin{aligned} \dot{x}_o &= V \cdot \cos\theta_o \\ \dot{y}_o &= V \cdot \sin\theta_o \\ \dot{\theta}_o &= \Omega \end{aligned} \quad \text{, donde} \quad \begin{aligned} V &= \frac{R}{2}\left(\omega_D + \omega_I\right) \\ \Omega &= \frac{R}{D}\left(\omega_D - \omega_I\right) \end{aligned} \qquad \text{<4.1>}$$

En la anterior ecuación 4.2, el sistema de la izquierda constituye específicamente el modelo inercial de la posición del robot, y el de la derecha las ecuaciones de cinemática diferencial inversa de la silla de ruedas.

[3] Se recuerda que se utiliza la notación subíndice 'o' para indicar que la información se obtiene por odometría, mientras que el subíndice 'v' para indicar que es información obtenida mediante el sistema de visión, tal y como se introdujo ya en el primer capítulo del documento.

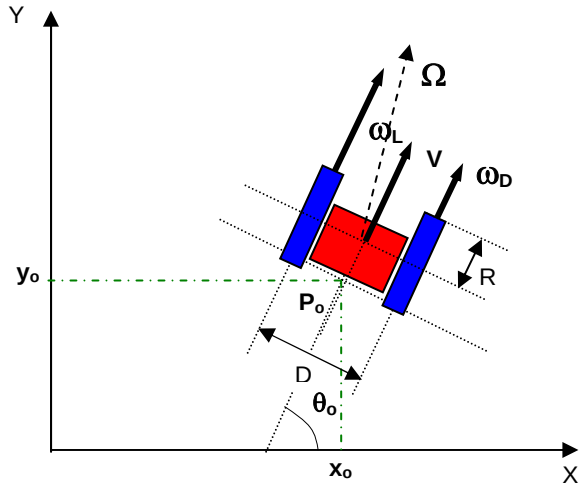


Figura 4.2. Diagrama explicativo de la cinemática diferencial del robot móvil de interés

El modelo obtenido se ha supuesto libre de ruidos, ya que estos serán caracterizados más adelante, mediante la correspondiente matriz de covarianza. Además, como el modelado se realiza con el objetivo de poder aplicarlo al estimador óptimo, deberá ser discreto, para poder implementarlo en el EKF.

A la vista del modelo matemático, se proponen como entradas del modelo discreto el vector de medidas de velocidad V y Ω (que se obtienen del sistema de odometría mediante la relación aritmética simple que se mostró en la ecuación <4.1>); como salidas el vector de posición en el plano de movimiento (XY) y la orientación del mismo respecto al eje X ($\vec{z}_{o,k} = \begin{bmatrix} x_{o,k} & y_{o,k} & \theta_{o,k} \end{bmatrix}^T$); y como estados del sistema, aquellos que determinan la evolución de la posición del mismo, y que coincide en este caso plenamente con las salidas elegidas ($\vec{x}_{o,k} = \begin{bmatrix} x_{o,k} & y_{o,k} & \theta_{o,k} \end{bmatrix}^T$). El diagrama de la figura 4.3 muestra la organización esquemática del modelo resultante.

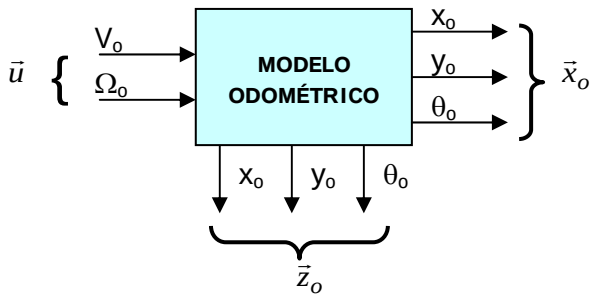


Figura 4.3.. Elección de las variables de interés del modelo odométrico del robot móvil

De este modo el modelo inercial lineal y discreto de la planta vendría descrito por las siguientes ecuaciones de estado y de salida:

$$\begin{bmatrix} x_{o,k} \\ y_{o,k} \\ \theta_{o,k} \end{bmatrix} = A \cdot \begin{bmatrix} x_{ok-1} \\ y_{o,k-1} \\ \theta_{ok-1} \end{bmatrix} + B \cdot \begin{bmatrix} V_k \\ \Omega_k \end{bmatrix}$$
$$\begin{bmatrix} x_{o,k} \\ y_{o,k} \\ \theta_{o,k} \end{bmatrix} = C \cdot \begin{bmatrix} x_{o,k-1} \\ y_{o,k-1} \\ \theta_{o,k-1} \end{bmatrix} + D \cdot \begin{bmatrix} V_k \\ \Omega_k \end{bmatrix} \qquad <4.2>$$

Donde se aprecia claramente que $D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$ y $C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

Sin embargo, comparando las ecuaciones <4.2> y <4.1> es también fácil de apreciar que no existe una relación lineal entre las componentes del vector de estado, por lo que no es posible obtener las matrices A y B del modelo. Es necesario, por tanto, hacer un estudio más exhaustivo del modelo odométrico.

4.1.1. Modelo discreto y linealizado del sistema odométrico

La descripción no lineal y continua del modelo odométrico queda, por tanto, tal y como se muestra en la ecuación <4.3>, y será necesario discretizarlo y, más tarde, linealizarlo para poder expresarlo con un formato como el mostrado en la ecuación <4.2>.

$$\begin{aligned} \dot{x}_o(t) &= V(t) \cdot \cos\theta_o(t) \\ \dot{y}_o(t) &= V(t) \cdot \sin\theta_o(t) \\ \dot{\theta}_o(t) &= \Omega(t) \end{aligned} \qquad <4.3>$$

En primer lugar, como se desea utilizar el modelo para estimar la posición en un procesador digital, éste se discretiza, dando lugar a las siguientes expresiones:

$$\begin{aligned} x_{o,k} &= x_{o,k-1} + T_s \cdot V_k \cdot \cos\theta_{o,k-1} \\ y_{o,k} &= y_{o,k-1} + T_s \cdot V_k \cdot \sin\theta_{o,k-1} \\ \theta_{o,k} &= \theta_{o,k-1} + T_s \cdot \Omega_k \end{aligned} \qquad <4.4>$$

Donde Ts es el periodo de muestreo o de procesado del modelo de interés. Se supone conocida la posición inicial del móvil (t=0 o k=0), por lo que la integración mostrada permite obtener la posición absoluta del robot en cualquier instante.

Sin embargo, para poder aplicar el filtrado de Kalman al modelo, es necesario conocer las matrices características del modelo cinemático discreto del sistema (ecuaciones <4.2>) por lo que se aplica la descomposición en serie de Taylor a la ecuación de estado presentada en <4.4>, y se obtiene el jacobiano

que constituirá la representación lineal de las matrices B y sobre todo A, que es necesaria para implementar el EKF. A la hora de aplicar la linealización por Taylor (idéntica a la presentada en la anterior ecuación <3.14>) se emplean únicamente el primer y segundo término de la serie, y se centran en el punto de trabajo del sistema[4].

$$f(\vec{x}_k) = f(\vec{x}_0) + A(\vec{x}_k - \vec{x}_0) + B(\vec{u}_k - \vec{u}_0) \text{ donde } \begin{array}{l} A \Rightarrow A_{[i,j]} = \dfrac{\partial f_{[i]}}{\partial \vec{x}_{[j]}}\left(\vec{x}_0, \vec{u}_0\right) \\ B \Rightarrow B_{[i,j]} = \dfrac{\partial f_{[i]}}{\partial \vec{u}_{[j]}}\left(\vec{x}_0, \vec{u}_0\right) \end{array} \quad \text{<4.5>}^{5}$$

Sin embargo, como el modelo linealizado (o mejor dicho sus matrices características) se van a emplear únicamente en el mundo discreto se obtiene el modelo linealizado y discretizado, a partir de la representación no lineal y discreta de la ecuación <4.4>, con el siguiente resultado:

$$A = \begin{bmatrix} \dfrac{\partial x_k}{\partial x_{k-1}}\left(x_0, \vec{u}_0\right) & \dfrac{\partial x_k}{\partial y_{k-1}}\left(y_0, \vec{u}_0\right) & \dfrac{\partial x_k}{\partial \theta_{k-1}}\left(\theta_0, \vec{u}_0\right) \\ \dfrac{\partial y_k}{\partial x_{k-1}}\left(x_0, \vec{u}_0\right) & \dfrac{\partial y_k}{\partial y_{k-1}}\left(y_0, \vec{u}_0\right) & \dfrac{\partial y_k}{\partial \theta_{k-1}}\left(\theta_0, \vec{u}_0\right) \\ \dfrac{\partial \theta_k}{\partial x_{k-1}}\left(x_0, \vec{u}_0\right) & \dfrac{\partial \theta_k}{\partial y_{k-1}}\left(y_0, \vec{u}_0\right) & \dfrac{\partial \theta_k}{\partial \theta_{k-1}}\left(\theta_0, \vec{u}_0\right) \end{bmatrix} = \begin{bmatrix} 1 & 0 & -V_0 \cdot T_s \cdot \sin(\theta_0) \\ 0 & 1 & V_0 \cdot T_s \cdot \cos(\theta_0) \\ 0 & 0 & 1 \end{bmatrix} \quad \text{<4.6>}$$

$$B = \begin{bmatrix} \dfrac{\partial x_k}{\partial V_k}\left(\vec{x}_0, V_0\right) & \dfrac{\partial x_k}{\partial \Omega_k}\left(\vec{x}_0, \Omega_0\right) \\ \dfrac{\partial y_k}{\partial V_k}\left(\vec{x}_0, V_0\right) & \dfrac{\partial y_k}{\partial \Omega_k}\left(\vec{x}_0, \Omega_0\right) \\ \dfrac{\partial \theta_k}{\partial V_k}\left(\vec{x}_0, V_0\right) & \dfrac{\partial \theta_k}{\partial \Omega_k}\left(\vec{x}_0, \Omega_0\right) \end{bmatrix} = \begin{bmatrix} T_s \cdot \cos(\theta_0) & 0 \\ T_s \cdot \sin(\theta_0) & 0 \\ 0 & T_s \end{bmatrix} \quad \text{<4.7>}$$

El modelo final en formato matricial queda:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ \theta_0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & -V_0 \cdot T_s \cdot \sin(\theta_0) \\ 0 & 1 & V_0 \cdot T_s \cdot \cos(\theta_0) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} - x_0 \\ y_{k-1} - y_0 \\ \theta_{k-1} - \theta_0 \end{bmatrix} + \begin{bmatrix} T_s \cdot \cos(\theta_0) & 0 \\ T_s \cdot \sin(\theta_0) & 0 \\ 0 & T_s \end{bmatrix} \begin{bmatrix} (V_k - V_0)^* \\ \Omega_k - \Omega_0 \end{bmatrix} \quad \text{<4.8>}$$

Y en formato de sistema de ecuaciones resulta:

[4] A partir de este punto, para simplificar la nomenclatura, se elimina la notación m_o (subíndice ‘o’) indicadora de que se trata del modelo odométrico del sistema móvil. Se recuperará cuando se obtenga el modelo completo.

[5] Se utilizará el subíndice ‘0’ para indicar que se trata del punto de linealización de la serie y no crear confusión con la notación temporal.

$$\begin{aligned} x_{o,k} &= x_{o,k-1} - V_0 \cdot T_s \cdot \sin(\theta_{o,0})(\theta_{o,k-1} - \theta_{o,0}) \\ y_{o,k} &= y_{o,k-1} + V_0 \cdot T_s \cdot \cos(\theta_{o,0})(\theta_{o,k-1} - \theta_{o,0}) \\ \theta_{o,k} &= \theta_{o,k-1} + T_s(\Omega_k - \Omega_0) \end{aligned} \qquad <4.9>$$

Planteadas todos los formatos del modelo odométrico sin ruido se pasa a presentar la caracterización del ruido que afecta a estas medidas. Sin embargo, antes de comenzar ese punto es necesario puntualizar acerca del uso de los diferentes modelos obtenidos. Teniendo en cuenta la aplicación final que se les va a dar (implementación de un EKF para fusionar medidas de posición), es fácil observar que el modelo no lineal y discreto será el utilizado para estimar el vector de estado en la etapa de predicción, mientras que las matrices del modelo discreto y linealizado (A_k y C_k, si bien esta última es constante e igual a la matriz identidad) serán utilizadas para implementar las ecuaciones de evolución del estimador de Kalman. En el capítulo siguiente se recopilará todo esto de forma resumida.

4.1.2. Modelo del ruido asociado al vector de estado

El punto que se acaba de presentar permite obtener (entre otros) un modelo no lineal y discreto del móvil. Tal y como se resumía en la introducción del capítulo este modelo odométrico permitirá obtener la estimación del vector de estado del sistema robótico mediante el uso de un filtrado de Kalman que elimine el ruido de medidas introducido por los sensores odométricos.

Es decir, el comportamiento dinámico del sistema, representado por la ecuación de estados $\vec{x}_k = f\left(\vec{x}_{k-1}, \vec{u}_k, \vec{w}_k\right)$, se estima sin tener en cuenta el ruido de medidas que afecta al estado, tal y como se contaba en el inicio de la definición del EKF (ver ecuación <3.13>), y mediante las ecuaciones <4.4> anteriores. Sin embargo, para modelar el comportamiento completo del sistema ha de incluirse la caracterización del ruido de medida $\vec{w}_k$ comentado.

Realmente, como lo que se desea es utilizar dicha representación para aplicarla al algoritmo del EKF es necesario linealizarla y presentarla de forma discreta, aplicando la ya mencionada ecuación <3.14> del mismo modo que se hizo con el modelo odométrico sin ruido en el apartado anterior, para llegar al final de la caracterización a una expresión del tipo:

$$f(\vec{x}_k) = \vec{x}_0 + A(\vec{x}_k - \vec{x}_0) + \ldots + W_k \cdot \vec{w}_k \qquad <4.10>$$ [6]

Donde $W_k \Rightarrow W_{[i,j]} = \dfrac{\partial f_{[i]}}{\partial \vec{w}_{[j]}}\left(\vec{x}_0, \vec{u}_0\right)$ (tal y como se presentó ya en la ecuación <3.17>), es la matriz que asocia el ruido asociado a la medida con el vector de estados.

El planteamiento parte, por tanto, del análisis del punto en el que se incorporan los ruidos de medida de los encoders:

[6] A partir de este punto se elimina la notación 'o', indicadora de que se está haciendo referencia al modelo odométrico, para simplificar las expresiones. Se recuperará a la hora de presentar la descripción final del mismo.

$$\begin{aligned} \omega_{In} &= \omega_I + w_I \\ \omega_{Dn} &= \omega_D + w_D \end{aligned} \qquad <4.11>$$

Aplicando esa nueva definición a la cinemática inversa del robot (ver ecuación <4.1>) resulta lo siguiente:

$$\begin{aligned} V_n &= \frac{R}{2}\left(\omega_{Dn} + \omega_{In}\right) = V + \frac{R}{2}\left(w_D + w_I\right) \\ \Omega_n &= \frac{R}{D}\left(\omega_{Dn} - \omega_{In}\right) = \Omega + \frac{R}{D}\left(w_D + w_I\right) \end{aligned} \qquad <4.12>$$

Con lo que el modelo no lineal discreto resultante queda, en este caso, como sigue (de la ecuación <4.4>):

$$\begin{aligned} x_{on,k} &= x_{o,k-1} + T_s \cdot V_{n,k} \cdot \cos\theta_{o,k-1} = x_{o,k-1} + T_s \cdot V_k \cdot \cos\theta_{o,k-1} + T_s \cdot \cos\theta_{o,k-1} \cdot \frac{R}{2}\left(w_D + w_I\right) = x_{o,k} + T_s \cdot \cos\theta_{o,k-1} \cdot \frac{R}{2}\left(w_D + w_I\right) \\ y_{on,k} &= y_{o,k-1} + T_s \cdot V_{n,k} \cdot \sin\theta_{o,k-1} = y_{o,k-1} + T_s \cdot V_k \cdot \sin\theta_{o,k-1} + T_s \cdot \sin\theta_{o,k-1} \cdot \frac{R}{2}\left(w_D + w_I\right) = y_{o,k} + T_s \cdot \sin\theta_{o,k-1} \cdot \frac{R}{2}\left(w_D + w_I\right) \\ \theta_{on,k} &= \theta_{o,k-1} + T_s \cdot \Omega_{n,k} = \theta_{o,k-1} + T_s \cdot \Omega_k + T_s \cdot \frac{R}{D}\left(w_D - w_I\right) = \theta_{o,k} + T_s \cdot \frac{R}{D}\left(w_D - w_I\right) \end{aligned}$$

<4.13>

A estos modelos se les aplica la linealización que se presentó en la ecuación <4.10>. En este caso se desea calcular la relación lineal entre el vector ruido $\vec{w}_k = \begin{bmatrix} w_D & w_I \end{bmatrix}^T$ y el vector de estado, $\vec{x}_k = \begin{bmatrix} x_{o,k} & y_{o,k} & \theta_{o,k} \end{bmatrix}^T$ aplicando la definición de W_k presentada más arriba, y obteniéndose los siguientes resultados:

$$W_k = \begin{bmatrix} \frac{\partial x_k}{\partial w_D}\left(\vec{x}_0, \vec{u}_0\right) & \frac{\partial x_k}{\partial w_I}\left(\vec{x}_0, \vec{u}_0\right) \\ \frac{\partial y_k}{\partial w_D}\left(\vec{x}_0, \vec{u}_0\right) & \frac{\partial y_k}{\partial w_I}\left(\vec{x}_0, \vec{u}_0\right) \\ \frac{\partial \theta_k}{\partial w_D}\left(\vec{x}_0, \vec{u}_0\right) & \frac{\partial \theta_k}{\partial w_I}\left(\vec{x}_0, \vec{u}_0\right) \end{bmatrix} = \begin{bmatrix} R/2 \cdot T_s \cdot \cos(\theta_0) & R/2 \cdot T_s \cdot \cos(\theta_0) \\ R/2 \cdot T_s \cdot \sin(\theta_0) & R/2 \cdot T_s \cdot \sin(\theta_0) \\ R/D \cdot T_s & -R/D \cdot T_s \end{bmatrix} \qquad <4.14>$$

Con esta matriz y las obtenidas en el apartado anterior (A y B) se podría presentar un modelo completo discreto linealizado con ruido del sistema odométrico. Sin embargo, teniendo en cuenta lo explicado en el capítulo anterior de este documento, queda claro que el ruido no se incluye en el modelo a utilizar en el filtrado sino en la obtención de la matriz de innovación del covarianza del error de estimación ($P_{k+1/k}$) en el desarrollo del EKF, que tal y como se presentaba ya en la figura 3.3, se calcula de la siguiente forma:

$$P_{k+1/k} = A_k \cdot P_{k/k} \cdot A_k^T + W_k \cdot Q \cdot W_k^T \qquad <4.15>$$

A la vista de esta ecuación se observa que se han obtenido ya todos los parámetros necesarios para implementar el filtrado de Kalman sobre el modelo odométrico, a falta únicamente, de la matriz de covarianza del error de estados Q y que tendrá un valor:

$$Q = E\left[\vec{w}_k \cdot \vec{w}_k^T\right] = \begin{bmatrix} \sigma^2_{w_D} & 0 \\ 0 & \sigma^2_{w_I} \end{bmatrix} \qquad <4.16>$$

El valor de esta matriz no se ha calculado en este trabajo, pero se obtiene, tal y como se explicaba en el capítulo anterior, mediante pruebas off-line que permitan realizar una estadística sobre el comportamiento ruidoso del modelo odométrico, con lo que el modelo de ruido quedaría:

$$f(\vec{x}_k) = f(\vec{x}_0) + A(\vec{x}_k - \vec{x}_0) + \ldots + W_k \cdot \vec{w}_k \text{, donde } W_k \cdot \vec{w}_k = \begin{bmatrix} R/2 \cdot T_s \cdot \cos(\theta_0) & R/2 \cdot T_s \cdot \cos(\theta_0) \\ R/2 \cdot T_s \cdot \sin(\theta_0) & R/2 \cdot T_s \cdot \sin(\theta_0) \\ R/D \cdot T_s & -R/D \cdot T_s \end{bmatrix} \cdot \begin{bmatrix} w_D \\ w_I \end{bmatrix}$$

<4.17>

Estando el vector de ruido $\vec{w}_k$ definido por sus estadísticos: covarianza $Q = \begin{bmatrix} \sigma^2_{w_D} & 0 \\ 0 & \sigma^2_{w_I} \end{bmatrix}$ y media nula.

Otro modelo este ruido se puede obtener a partir de los diferentes trabajos que han sido realizados a propósito del tema del ruido producido por los modelos de posicionamiento por dead-reckoning, de entre los que merece la pena destacar el de [Borenstein&Feng-95]. Ellos dividen los ruidos de odometría en sistemáticos y no sistemáticos. Los primeros se deben a errores en el modelo mecánico del móvil (que, según sus investigaciones, se pueden resumir fundamentalmente en dos: asimetría del radio de las ruedas y desconocimiento de la base de las ruedas, y por tanto de la separación entre ruedas), y han de ser eliminados pues en caso contrario producirían un ruido de media no nula (no válido para la aplicación del EKF). Para ello, Borenstein y Feng proponen un test (UMBmark) que permite cuantificar dicho error y obtener unas constantes de corrección que evitan este ruido añadidas a las ecuaciones dinámicas del modelo. El test consiste en hacer que el robot recorra un circuito cuadrado varias veces en ambos sentidos para computar posteriormente el error de posicionamiento final que se produce en cada recorrido (ver figura 4.4). El valor medio del error total cometido permite calcular los factores de corrección (E_b y E_d) a aplicar a la cinemática del robot.

Este experimento fue realizado sobre el robot de interés en este trabajo [Moreno-02], obteniendo los valores de calibración que aseguran la eliminación del error sistemático en la plataforma de desarrollo, y por tanto anulan el valor medio estadístico del vector $\vec{w}_k$. Para obtener las ecuaciones de calibración, en primer lugar se obtiene el error relativo en las coordenadas (x_i,y_i) de cada recorrido realizado, y se calcula el centro de gravedad del circulo de error ($X_{cw,ccw}$,$Y_{cw,ccw}$), obteniéndose los siguientes resultados:

$$X_{cw,ccw} = \frac{1}{n}\sum_{i=1}^{n} x_i \Rightarrow \begin{matrix} 535{,}2mm(cw) \\ 433{,}2mm(ccw) \end{matrix} \quad , \quad Y_{cw,ccw} = \frac{1}{n}\sum_{i=1}^{n} y_i \Rightarrow \begin{matrix} 87{,}8mm(cw) \\ -138{,}4mm(ccw) \end{matrix} \qquad <4.18>$$

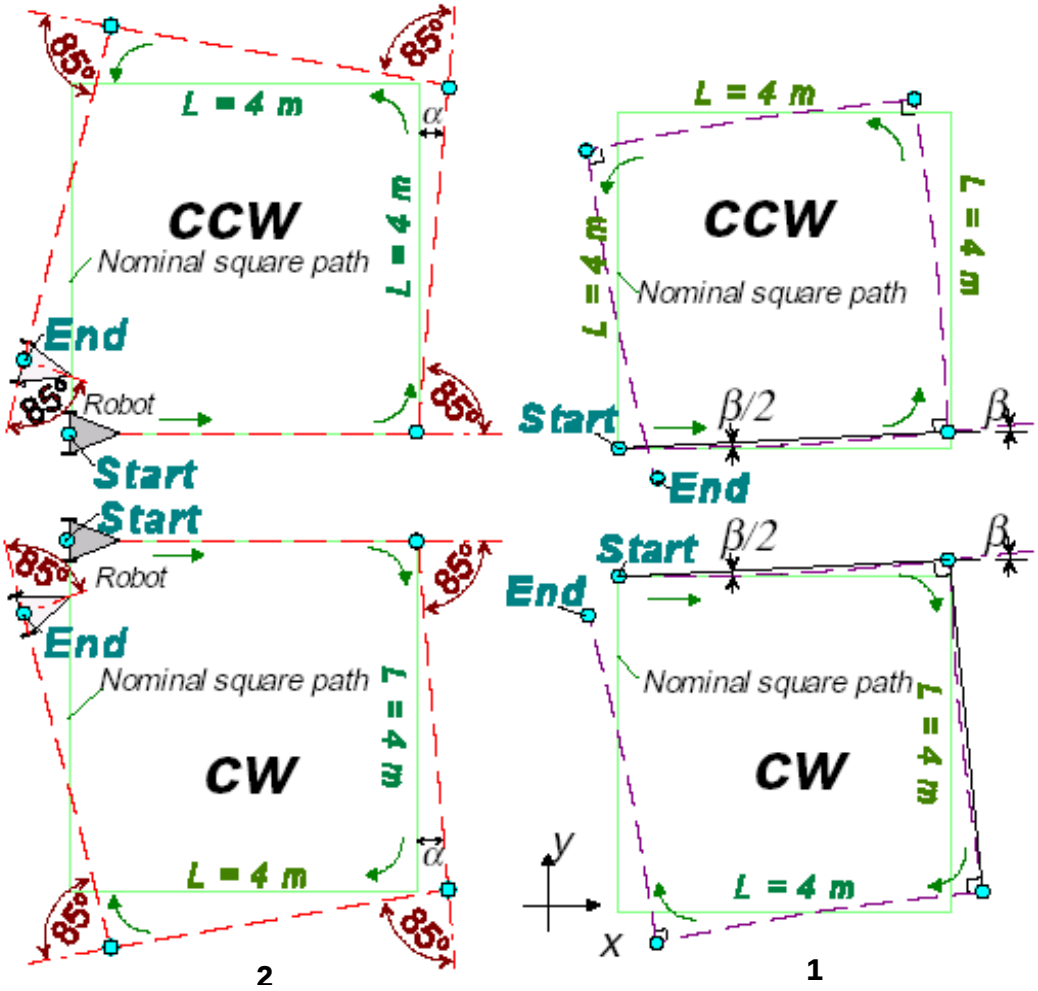


Figura 4.4. Imagen de ejemplo de la ejecución del algoritmo UMBmark en los dos sentidos de recorrido (CCW y CW) y el efecto de los factores principales (1: asimetría en el radio de las ruedas E_d y 2: incertidumbre en la separación entre ruedas E_b) en el error de posicionamiento final

Donde ‘n’ representa a las 5 veces que se realiza el recorrido en cada sentido, y ‘L’ la longitud en metros de los lados de cuadrado a recorrer (ver figura 4.4). A continuación se calculan los factores de corrección de odometría de la siguiente forma:

- **Caso 1 de la figura 4.4.** Con respecto a la diferencia entre los radios de las ruedas. Se produce un error de posición que depende del sentido de giro. El factor de calibración se calcula de este modo:

$$\beta = \frac{X_{cw} - X_{ccw}}{-4L} \cdot 180/\pi = -0{,}365^{o}$$

con lo que $E_d = \dfrac{K + D/2}{K - D/2} = 1{,}0005655$ donde $K = \dfrac{L}{2 \cdot \sin(\beta/2)}$ <4.19>

Este factor de corrección ha de aplicarse al valor del radio de las ruedas, o en su defecto sobre la lectura recogida de los encoders de realimentación de posición, de este modo:

$$E_d = \frac{R_D}{R_I} \Rightarrow R_D = E_d \cdot R_I \Rightarrow \omega_{D,calibrada} = Ed \cdot \omega_D \qquad <4.20>$$

Tal y como se puede apreciar en el valor obtenido para este factor de corrección, su efecto no va a ser prácticamente apreciable, sobre todo debido a que se perderá por cuantificación en función del formato de numeración empleado. Además hay que tener en cuenta que esta calibración no es todo lo robusta que se desearía, ya que la asimetría de los radios de las ruedas del robot cambia constantemente debido a diversos factores (estado de inflado de las mismas, distribución del peso en la silla, etc.). Por lo que la calibración no va a ser todo lo exacta sería deseable, tal y como se observará más adelante, sería necesario realizarla periódicamente con una batería de datos recogida on-line en el movimiento del robot.

- **Caso 2 de la figura 4.4.** Con respecto a la incertidumbre de la distancia entre ruedas. Se produce un error de orientación en cada giro del móvil, independientemente de la dirección de mismo. El factor de corrección se calcula del siguiente modo:

$$\alpha = \frac{X_{ccw} + X_{cw}}{-4L} \cdot 180/\pi = -3{,}46^{o} \text{ y } E_b = \frac{90^{o}}{90^{o} - \alpha} = 1{,}03998 \qquad <4.21>$$

Con lo que el valor calibrado a utilizar como distancia entre ruedas en las ecuaciones de cinemática debe ser:

$$D_{calibrado} = D \cdot E_b = D \cdot 1{,}03998 \qquad <4.22>$$

En lo que al segundo tipo de errores se refiere (errores no sistemáticos), que además han de ser modelados para tenerlos en cuenta en el proceso de filtrado, los mismos investigadores diseñan otro método de caracterización del error, semejante al anterior y denominado UMBmark Extendido. En este caso se pretende cuantificar el error cometido por la odometría directamente sobre el vector de estados $\vec{w}_k = \begin{bmatrix} w_x & w_y & w_z \end{bmatrix}^T$, forzando una situación propicia para este tipo de ruidos, a saber, introduciendo obstáculos cilíndricos de pequeño tamaño en el primer avance del robot (ver recorrido de la figura 4.6). A continuación el proceso es idéntico al del test estándar: procesar el error de posicionamiento final de forma estadística para hallar su varianza. Sin embargo ya de partida los autores comentan que este método solamente es útil para comparar el comportamiento de diferentes robots ante la misma situación (tipo de suelo y estado del mismo) y aún así dependerá del estado de las ruedas motrices del móvil.

Aplicando este hecho al objetivo de modelado del ruido se puede concluir asegurando que no hay modo de obtener un dato estadístico que modele estáticamente y de forma robusta el valor del ruido

asociado al dead-reckoning. La única solución es la ya comentada de realizar pruebas de movimiento off-line en unas condiciones lo más parecidas posibles a las que el robot se encuentre en su funcionamiento natural, para computar a partir de ellas el valor de varianza de posicionamiento obtenido en dicho movimiento.

Con este objetivo, el UMBmark simple realizado sobre la silla de ruedas automatizada para llevar a cabo el proceso de calibración antes descrito, fue utilizado también para obtener un modelo off-line y estático del error no sistemático de odometría. Para lograr este objetivo se realiza de nuevo el test una vez corregidos los problemas de calibración ya presentados (así la variable aleatoria a caracterizar $\vec{w}_k$ será, supuestamente, de media nula). El procedimiento consiste en computar el error cometido sobre la posición final en cada uno de los movimientos, después de hacer que el robot recorra el circuito 5 veces en cada sentido (ccw y cw) y hallar la matriz de covarianza de este error de posicionamiento global. El resultado gráfico del experimento se muestra en la figura 4.5.

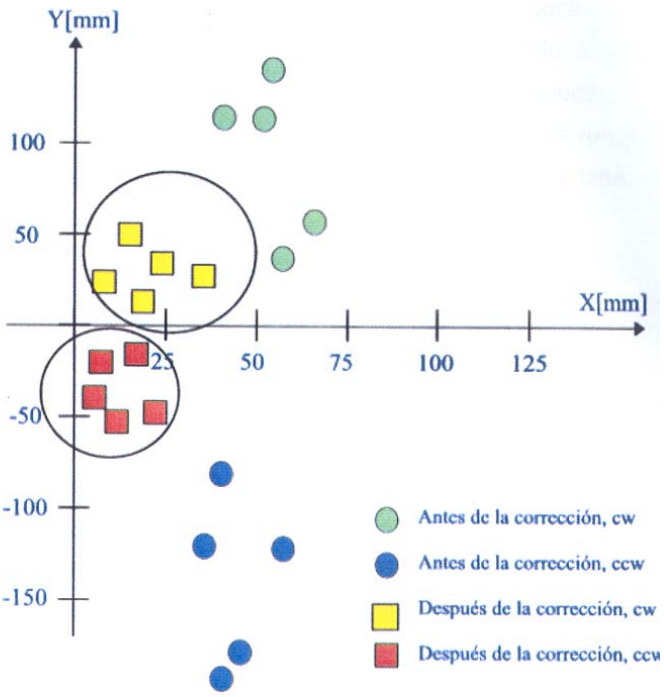


Figura 4.5. Resultado de los test UMBmark sobre el móvil de la aplicación antes y después del proceso de calibración

A partir de los resultados de error en el par de coordenadas (x,y) para cada uno de los 10 recorridos realizados se obtiene el valor de la varianza estadística de lo que sería el sub-vector de error $\vec{w}_k = \begin{bmatrix} w_x & w_y \end{bmatrix}^T$.

$$Q_1 = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} = \begin{bmatrix} 0.000438 & 0 \\ 0 & 0.0014 \end{bmatrix} \qquad <4.23>$$

Tal y como se observa en la expresión anterior, se ha nombrado a la matriz de covarianza como Q_1 puesto que sólo incluye parte de la matriz Q total. Falta por introducir el valor de la varianza de error de orientación σ_θ^2. Además, en el resultado mostrado en la anterior ecuación <3.28> aparece claramente el hecho de que los errores en las diferentes variables del vector de estado están incorrelados, tal se esperaba de estos tipos de ruido.

Para calcular el último elemento de la matriz de covarianza total Q, es necesario realizar otro planteamiento. En este caso se utiliza el test UMBmark Extendido, ya que en este test el error de orientación final del robot es insensible al sentido del recorrido realizado y a la localización de los obstáculos en él. La figura 4.6 muestra el recorrido utilizado para implementar esta tercera prueba. Se observa que se sitúan los obstáculos (10 pequeños cables cilíndricos de 10mm de grosor) al final del primer tramo del recorrido cuadrado.

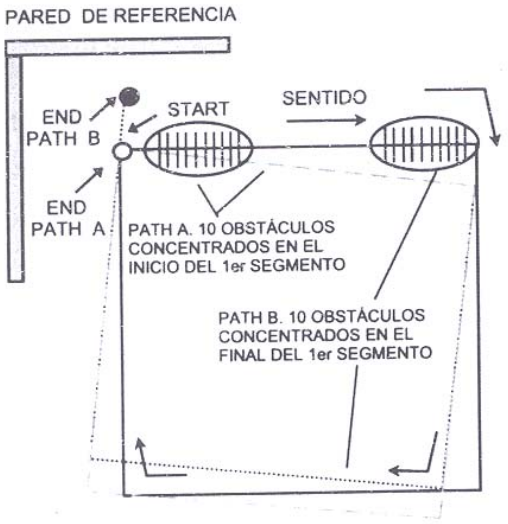


Figura 4.6. Diagrama explicativo del recorrido implicado en el test UMBmark extendido

El experimento consiste en computar el error de orientación del robot después de realizar cada uno de los 5 recorridos en los dos sentidos, y obtener la varianza de dicho error. Se supone de nuevo que la variable aleatoria que se desea caracterizar tiene media nula, puesto que se ha realizado previamente la calibración del robot, si bien esta suposición no es del todo correcta, por las razonas ya comentadas.

El valor final de la varianza buscada es de $\sigma_\theta^2 = 0.00030625$, con lo que la matriz de covarianza Q finalmente resulta:

$$Q = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix} = \begin{bmatrix} 0.000438 & 0 & 0 \\ 0 & 0.0014 & 0 \\ 0 & 0 & 0.00030625 \end{bmatrix} \qquad <4.24>$$

Es interesante destacar el hecho de que los resultados de varianza obtenidos coinciden más o menos en rango de magnitud con los presentados por los autores del test UMBmark correspondientes a robots de cinemática semejante, lo cual valida en parte estos resultados.

Otro aspecto que merece la pena comentar es el hecho de que todos los experimentos han sido realizados contando con encoders pasivos (o no deslizantes) en el móvil bajo estudio. Estos dispositivos permiten obtener un valor más o menos estático de Q, ya que los deslizamientos (la fuente de errores no sistemáticos más importante en el posicionamiento por odometría del robot y en recorridos interiores) harían que esta matriz variase en cada tramo recorrido, no permitiendo obtener un valor que reflejase su realidad general.

Finalmente, merece la pena destacar de la figura 4.5 anterior, que tal y como se presentía, la calibración no ha sido del todo correcta, puesto que el círculo o elipse de error no está centrado en el origen. Este efecto quita validez al modelado del vector de ruido $\vec{w}_k$, pero en cualquier caso es válido para incorporarlo al EKF, que en cualquier caso convergerá, tal y como se demuestra en el siguiente capítulo. Hay que tener en cuenta que, incluso así, la matriz Q obtenida tampoco es del todo válida, puesto que aquí se ha supuesto estática y por diversas razones ya comentadas debería ser tratada como dinámica.

Una solución más completa consistiría, por tanto, en actualizar el modelo de este tipo de errores on-line, es decir, obtener el valor de Q_k de forma dinámica, para incorporarlo así al EKF. Para ello sería necesario calcular el ruido de odometría comparando la posición obtenida por el modelo con la estimada a posteriori ($x_{k+1/k+1}$) por el filtro e ir actualizando así el valor de la matriz de covarianza. Sin embargo esta solución tiene dos inconvenientes:

1. El ruido de estados modelado dependería de la estimación del sistema, lo cual significaría que estos dos procesos estarían correlados. Esto choca con una de las premisas más importante que han de cumplirse para que el algoritmo recursivo de Kalman pueda converger.

2. El tiempo de proceso aumentaría en el algoritmo final de posicionamiento, sin aportar mucha mejora, pues, tal y como ya se ha comentado, en el capítulo siguiente se presentarán las simulaciones demuestran que el filtro funciona aún con cierto desconocimiento sobre el modelo del ruido.

Otra solución se encuentra ojeando los trabajos de otro autor que critica el método UMBmark aquí utilizado. Se trata de un estudio realizado por Kleeman y Chong que establece un nuevo modelo de dead-reckoning para el robot, mediante la definición de un nuevo vector de entrada $\vec{u}_k = \begin{bmatrix} L_k & R_k \end{bmatrix}^T$ (figura 4.7). Según los resultados mostrados en el artículo [Kleeman&Chong-96], este nuevo método proporciona un modelo de ruido consistente, cosa que no ocurría con el anterior, si bien cambia ligeramente en función del tipo de trayecto realizado por el robot en cada momento (curva simple, recta o giro sobre sí mismo), ya que las entradas se ven muy afectadas por la forma del camino. Esta técnica se propone como trabajo futuro, con el fin de obtener un modelo que represente con mayor fidelidad al sistema bajo estudio.

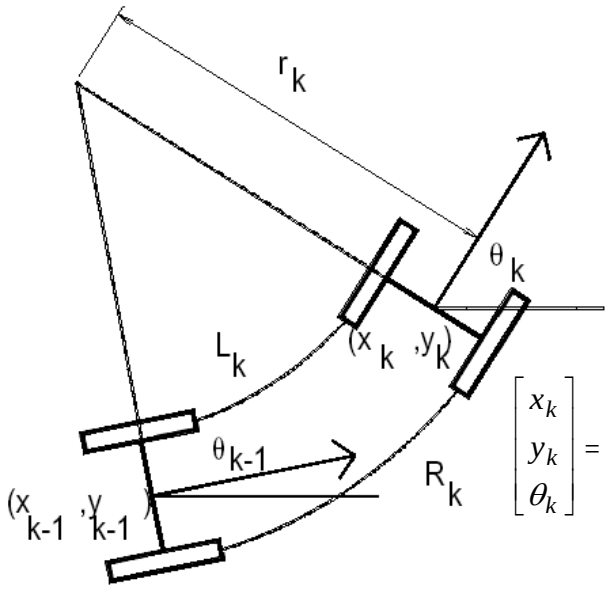
Como conclusión se puede decir que a raíz de todo el desarrollo, se ha obtenido un modelo estático de la matriz de covarianza del error de estados Q. Sin embargo, tal y como se ha podido observar en el planteamiento, esta matriz incluye valores de varianza directamente relacionados con las variables del vector

de estados, por lo que la matriz de asociación W_k ha de ser en este caso igual a la matriz identidad. Es decir, el nuevo modelo del ruido de estados será:

$$f(\vec{x}_k) = f(\vec{x}_0) + A(\vec{x}_k - \vec{x}_0) + \ldots + W_k \cdot \vec{w}_k \text{, donde } W_k \cdot \vec{w}_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} w_x \\ w_y \\ w_\theta \end{bmatrix} \qquad <4.25>$$

Estando el vector $\vec{w}_k$ definido por sus estadísticos de media nula y covarianza:

$$Q = \begin{bmatrix} 0.000438 & 0 & 0 \\ 0 & 0.0014 & 0 \\ 0 & 0 & 0.00030625 \end{bmatrix}$$



$$r_k = \frac{D}{2}\left(\frac{L_k + R_k}{L_k - R_k}\right)$$

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} r_k\left[\sin(\theta_{k-1}) - \sin(\theta_{k-1} + \frac{R_k - L_k}{D})\right] \\ r_k\left[\cos(\theta_{k-1} + \frac{R_k - L_k}{D}) - \cos(\theta_{k-1})\right] \\ \frac{R_k - L_k}{D} \end{bmatrix}$$

Figura 4.7. Nuevo modelo cinemático del móvil propuesto por Kleeman y Chong [Kleeman&Chong-96]

4.2. Modelo del sistema de visión

Como ya se comentó al principio del documento, un sistema de posicionamiento por visión basado en el reconocimiento y caracterización de marcas artificiales en el entorno es el segundo de los elementos que proporcionan información sobre la posición del robot (ver figura 4.1 anterior). Será necesario obtener un modelo de este segundo sistema para poder fusionar la información que proporciona con la suministrada por el modelo odométrico.

El sistema de posicionamiento local (SPL) diseñado por el Dr. D. Juan Carlos García García en su tesis doctoral [García-01], se basa en la detección y procesamiento por visión artificial de unas marcas artificiales especiales que se encuentran localizadas estratégicamente en el entorno de movimiento del

robot. Los algoritmos desarrollados en su trabajo permiten, una vez localizada la marca en la imagen del entorno, obtener la posición relativa de la cámara (y por lo tanto del móvil, ya que la anterior va solidaria a éste) con respecto a la marca. En la figura 4.8 se muestra dicha relación (vector V1 en la figura), así como un gráfico esquematizado de la marca artificial específicamente diseñada para el algoritmo SPL.

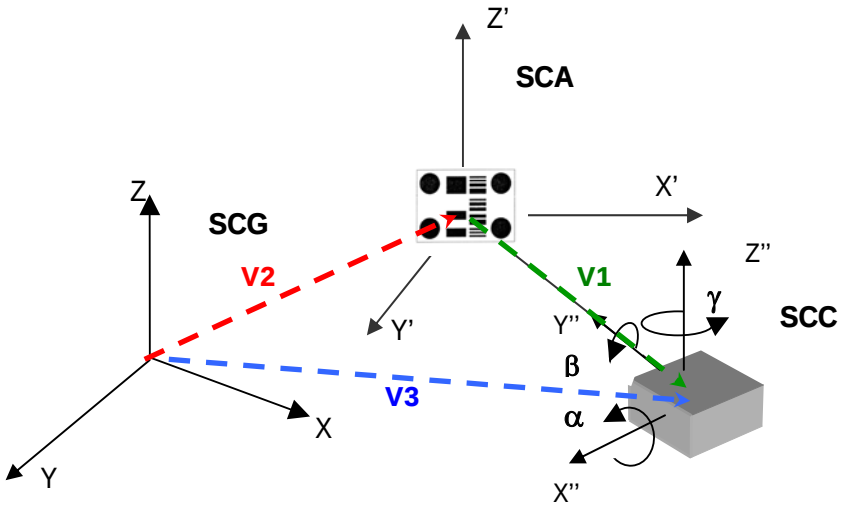


Figura.4.8. Representación espacial de los sistemas de coordenadas global del entorno (SCG XYZ), local de la marca (SCA, X'Y'Z') y de la cámara (SCC X''Y''Z''). Vectores de relación entre los tres sistemas:

SCC-SCA en verde
SCG-SCA en rojo
SCG-SCC en azul

Tal y como se mostraba en la ya mencionada figura 4.1, la salida del algoritmo SPL se utiliza para obtener, mediante una transformación simple, la posición absoluta del robot en el entorno de movimiento ($\vec{z}_{v,k}$), que será realmente la salida final del modelo de visión utilizado en el algoritmo EKF. Para poder realizar esta transformación, el algoritmo SPL posee una segunda funcionalidad, ya que la marca artificial diseñada al efecto incluye un código de barras que la identifica unívocamente dentro del entorno de movimiento, y a través del cual se obtiene su posición absoluta en el entorno (vector V2 en la figura 4.8), y con ella la del robot (vector V3 en la figura 4.8), gracias a un mapa del medio generado off-line (base de datos) que ha de poseer el sistema de visión para realizar el posicionamiento absoluto. La figura 4.9 muestra esta segunda funcionalidad.

El sistema de navegación realizado en el trabajo [Marrón-00] utiliza además estas marcas artificiales como nodos en la planificación de rutas y de trayectorias, tanto es así que el mapa comentado no sólo incluye información acerca de la posición absoluta de cada marca sino también ciertos datos interesantes para el software de navegación a la hora de diseñar los caminos que ha de pasar por el nodo.

Como ya se comentó en anteriores ocasiones, el papel del modelo del sistema de visión en la implementación del EKF se reduce a proporcionar el valor del vector de salida de posición sensada en cada momento ($\vec{z}_{v,k} = \begin{bmatrix} x_{v,k} & y_{v,k} & \theta_{v,k} \end{bmatrix}^T = \vec{x}_{v,k}$). Esta información coincide totalmente con la salida de

posición global del algoritmo de visión ya comentada, por lo que no será necesario implementar ningún modelo específico de este sistema para incorporar la información sensada al EKF, sino simplemente aplicar la correspondiente transformación a la salida de posición local proporcionada por el SPL.

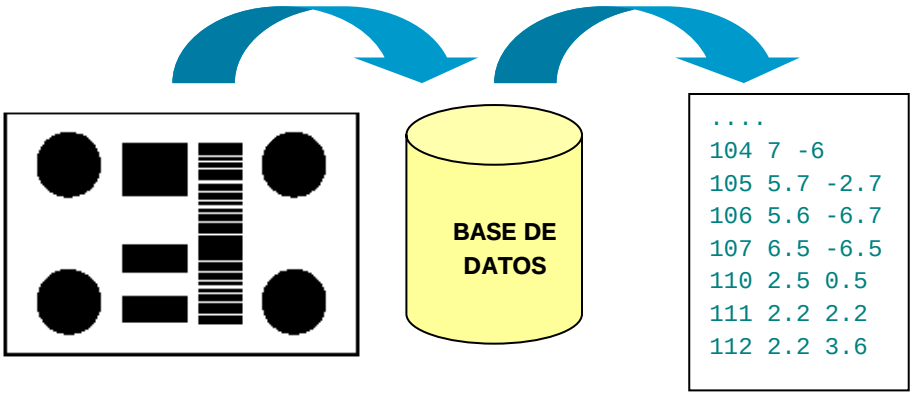


Figura 4.9. Esquema ilustrativo de la obtención de la posición absoluta de la marca en el SCG mediante el código de barras

El sistema de visión constituirá, por tanto el elemento sensorial de la salida en el algoritmo de fusión, que se presentó en el capítulo anterior, por la siguiente ecuación:

$$\vec{z}_k = f\left(\vec{x}_k, \vec{v}_k\right), \quad \text{o directamente} \quad \vec{z}_k = \vec{x}_k + V_k \cdot \vec{v}_k \qquad <4.26>^{7}$$

Además, tal y como se observa en la ecuación anterior, será necesario cuantificar mediante la consiguiente matriz de covarianza (R) y de relación (V_k) el ruido asociado a esta nueva medida. La transformación y las caracterizaciones mencionadas son las que se van a abordar a continuación.

4.2.1. Modelo de posición absoluta a partir del algoritmo SPL

En la figura 4.10 se muestra de nuevo el funcionamiento del sistema de posicionamiento por visión (SPL) incorporado en la silla de ruedas. Como ya se ha comentado, enmarcado en el trabajo de tesis del Dr. D. Juan Carlos García García, se desarrolló el algoritmo de visión que permite obtener la posición del móvil relativa a la marca $\vec{z}'_{v,k} = \begin{bmatrix} x'_{v,k} & y'_{v,k} & \gamma_{v,k} \end{bmatrix}^T$, donde $\vec{z}'_{v,k} = h\left(\vec{x}'_{v,k}, 0\right) = \vec{x}'_{v,k}$ (es decir suponiendo nulo el ruido de las medidas), y donde el superíndice prima(') se ha añadido para indicar que se trata de medidas de posición del robot relativas respecto a la cámara (al sistema de coordenadas SCA) y no al sistema de posicionamiento global (SCG).

[7] Se ha planteado esta simplificación teniendo en cuenta que la salida coincide con el vector de estados en este sistema y que el ruido y la salida se hacen independientes a través de la correspondiente matriz de asociación V_k, que no ha de confundirse con la entrada de velocidad del modelo odométrico.

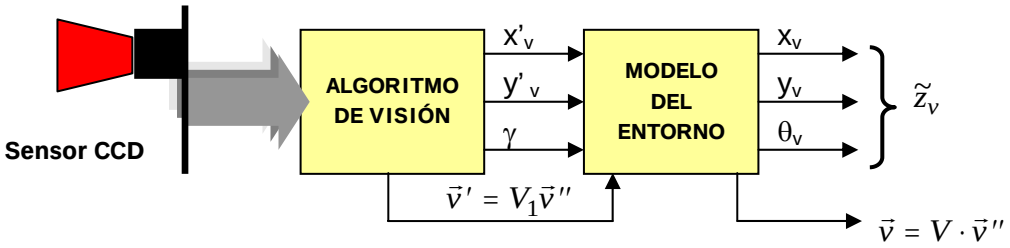


Figura 4.10. Gráfico esquemático de la obtención de $\vec{z}_{v,k}$ a partir del SPL ($\vec{z}'_{v,k}$)

En la tesis se desarrollan dos métodos SPL diferentes para poder obtener el vector $\vec{z}'_{v,k}$ a partir de la localización en la imagen de los centroides de los cuatro círculos que incluye la marca (esta se puede observar en la figura 4.9 anterior). La primera técnica permite obtener sin ningún tipo de información previa el vector de traslación ($\vec{T}_k = \begin{bmatrix} x'_{0,k} & y'_{0,k} & z'_{0,k} \end{bmatrix}^T$) y el de rotación completo ($\vec{\Omega}_k = \begin{bmatrix} \alpha_{0,k} & \beta_{0,k} & \gamma_{0,k} \end{bmatrix}^T$ [8]) del SCC con respecto al SCA (que se corresponden con el vector V1 de la figura 4.8 anterior) mediante un método de análisis de puntos de fuga. Con todo ello, el vector buscado se obtiene a partir de los anteriores de la siguiente forma:

$$\vec{z}'_{v,k} = \begin{bmatrix} x'_{v,k} & y'_{v,k} & \gamma_k \end{bmatrix}^T = \begin{bmatrix} x'_{0,k} & y'_{0,k} & \gamma_{0,k} \end{bmatrix}^T \qquad <4.27>$$

El segundo algoritmo SPL desarrollado en la tesis presupone conocido parte del vector de orientación y obtiene los datos de salida $\vec{z}'_{v,k}$ por simple proyección tridimensional. Las suposiciones realizadas en este último método se hacen teniendo en cuenta que la marca estará localizada siempre en un plano (X'Z') perpendicular al de rodadura (X'Y'). Con ello se puede considerar:

1. β nulo debido a que el plano de proyección de la imagen (X'Z') es perpendicular al plano de rodadura (X'Y').

2. α conocido pues ha de existir un pant-tilt controlado a bordo del robot que permita orientar el eje de proyección de la cámara. Las pruebas incluidas en la tesis se han realizado con α=90º, es decir, eje de proyección (z') paralelo al plano de rodadura (X'Y').

Los resultados obtenidos son mucho mejores con este segundo método (método simplificado), sobre todo en lo que a sensibilidad del algoritmo con respecto a γ se refiere, no siendo aceptables los obtenidos con el primer método, sobre todo si se tiene en cuenta la importancia de esta variable en las tareas de navegación. En ambos casos, en la recuperación del vector $\vec{z}'_{v,k}$ se incorpora un algoritmo de mínimos cuadrados que minimice la incorporación de ruidos en los resultados del SPL.

[8] No confunda este vector con la entrada de velocidad de giro del modelo odométrico.

El proceso completo incluye, en total, una etapa de búsqueda de un patrón que identifique a la marca dentro de la imagen (se usa específicamente un código Barker de 7 dígitos); una de obtención del mencionado vector de posición $\vec{z}'_{v,k}$ mediante la técnica considerada (en general la segunda); y un último proceso de reconocimiento del código de barras incluido en la marca, que permite localizarla en el mapa del entono que posee el sistema de visión, y obtener a partir de éste la posición absoluta de la marca en el sistema de coordenadas global (el vector V2 de figura 4.8).

Conocidos los vectores V1 y V2 de la figura 4.8, el vector V3, que es la posición absoluta del robot $\vec{z}_{v,k}$ (salida del modelo de visión a incorporar al EKF) se obtiene fácilmente mediante las siguientes relaciones trigonométricas (ver figura 4.11):

$$\begin{aligned} \vec{V}3 &= \vec{V}1 + \vec{V}2 \\ x_{v,k} &= x_{m,i} + x'_{0,k} \cdot \cos(\theta_{m,i}) - y'_{0,k} \cdot \sin(\theta_{m,i}) = x_{m,i} + x'_{v,k} \cdot \cos(\theta_{m,i}) - y'_{v,k} \cdot \sin(\theta_{m,i}) \\ y_{v,k} &= y_{m,i} + x'_{0,k} \cdot \sin(\theta_{m,i}) + y'_{0,k} \cdot \cos(\theta_{m,i}) = y_{m,i} + x'_{v,k} \cdot \sin(\theta_{m,i}) + y'_{v,k} \cdot \cos(\theta_{m,i}) \\ \theta_{v,k} &= \theta_{m,i} + \theta'_{v,k} = \theta_{m,i} + \gamma_k \end{aligned} \qquad <4.28>$$

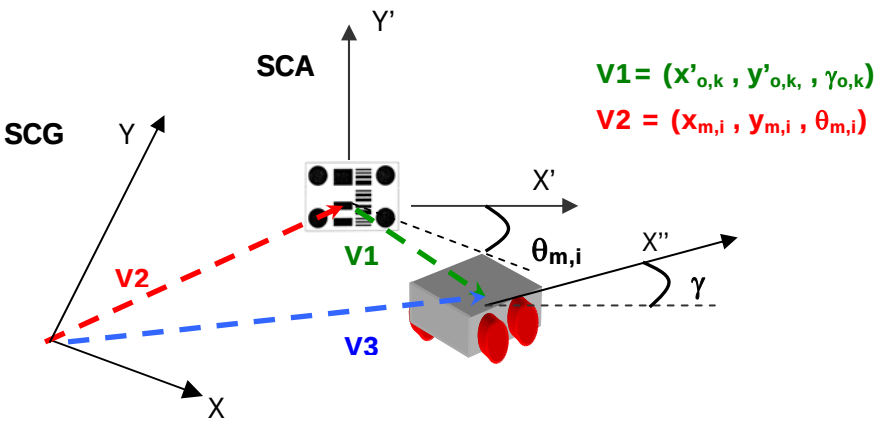


Figura 4.11. Representación 2D de la transformación a aplicar a la salida del algoritmo SPL ($\vec{z}'_{v,k}$) para obtener la posición absoluta del móvil (salida del algoritmo global de visión $\vec{z}_{v,k}$)

De la figura 4.11 y la anterior expresión cabe destacar el valor de las componentes del vector V2: $P_{m,i} = \begin{bmatrix} x_{m,i} & y_{m,i} & \theta_{m,i} \end{bmatrix}^T$ que se obtienen, como ya se comentó, del mapa del entorno, y cuya notación 'i' identifica a la marca en cuestión con respecto a la que el algoritmo SPL haya proporcionado la salida de posición.

Tal y como se observa en la figura 4.11 el problema termina siendo únicamente una relación 2D, ya que no se tiene en cuenta la coordenada 'z' del sistema de coordenadas global. Aunque ésta se podría obtener perfectamente con los mencionados algoritmos SPL, y completar con ella el vector de

posicionamiento total ($\vec{z}_{totalv,k} = \begin{bmatrix} x_{v,k} & y_{v,k} & z_{v,k} & \theta_{v,k} \end{bmatrix}^T$), no se considera necesaria teniendo en cuenta los objetivos de posicionamiento planteados en este trabajo.

4.2.2. Modelo del ruido asociado al vector de salida

En los algoritmos SPL desarrollados en la tesis mencionada, se realiza además un estudio detallado para analizar los errores que incorporan los resultados de posición debido a circunstancias diversas (fundamentalmente iluminación ambiente, distancia móvil-marca, orientación de la marca, etc.). Todas estas causas se agrupan y se caracterizan por lo que se ha dado en llamar "varianza del error de píxel".

Esta varianza se extrae del error obtenido en el cálculo de los centroides de los círculos que aparecen en la marca artificial, y que son la base para obtener la posición local del robot.

Para llevar a cabo el análisis, el autor de la tesis realizó experimentos con un banco de pruebas de 300 imágenes capturadas en 30 puntos diferentes del entorno de la marca (10 imágenes por punto de prueba), organizados a distintas distancias (de 1 a 5m con incrementos de 1m entre puntos de prueba) y orientaciones (de 0 a -75º de orientación del eje de proyección de la cámara con respecto a la marca, con incrementos entre puntos de prueba de -15º). Las imágenes se capturaron con muy distintas características de iluminación y con un fondo no preparado, por lo que representan fielmente la situación del algoritmo SPL en funcionamiento normal.

Con estas condiciones se procesa el algoritmo simplificado sobre todas las imágenes y se obtiene un error de píxel de varianza muy semejante en todos los puntos de prueba y de un valor de alrededor de los 0.011 pixel^2.

Además el autor también realiza un estudio sobre la propagación de éste a la salida de posición de su algoritmo ($\vec{z}'_{v,k}$), lo cual es de gran utilidad en este trabajo para incorporar la información de ruido al EKF. Concretamente, el análisis permite obtener la matriz de covarianza R asociada a un vector transformado (en coordenadas polares) que va a permitir analizar de forma más gráfica las elipses de error del algoritmo SPL. El vector utilizado para caracterizar el ruido es el siguiente:

$$\vec{z}''_{v,k} = \begin{bmatrix} r_k & \sin(\gamma_k) & \cos(\gamma_k) \end{bmatrix}^T \qquad <4.29>$$

Donde r_k es el promediado de distancia del robot a la marca, sobre el eje óptico de la cámara (SCC). Para poder relacionar este vector $\vec{z}''_{v,k}$ con el vector $\vec{z}'_{v,k}$ es necesario aplicar las ecuaciones de transformación tridimensional correspondientes, si bien aplicando las condiciones expuestas en el método simplificado (β=0 y α conocido) la relación no lineal existente entre los dos vectores puede obtenerse mediante sencillas transformaciones tridimensionales, tal y como se muestra en la figura 4.12.

A partir de la figura comentada, es fácil obtener las relaciones buscadas, que vendrán dadas por las expresiones que se muestra a continuación:

$$x'_{v,k} = r_k \cdot \cos(\gamma_k - 270) = -r_k \cdot \sin(\gamma_k)$$
$$y'_{v,k} = r_k \cdot \sin(\gamma_k - 270) = r_k \cdot \cos(\gamma_k) \qquad <4.30>$$
$$\theta'_{v,k} = \gamma_k$$

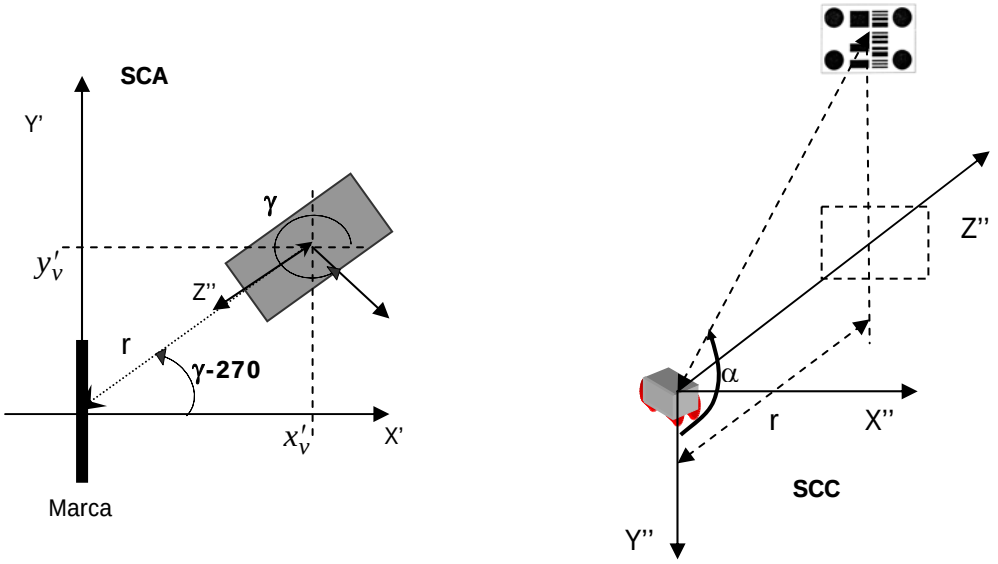


Figura 4.12. Transformación inversa del vector de coordenadas polares de caracterización del ruido en el SPL

Además es necesario aplicar a estos resultados la transformación SCC-SCG que fue expuesta en la anterior ecuación <4.28>, para relacionar finalmente el vector de ruido $\vec{v}_k = \begin{bmatrix} v_r & v_{\sin\gamma} & v_{\cos\gamma} \end{bmatrix}^T$ con el de salida de posición $\vec{z}_{v,k}$:

$$x_{v,k} = x_{m,i} + x'_{v,k} \cdot \cos(\theta_{m,i}) - y'_{v,k} \cdot \sin(\theta_{m,i}) = x_{m,i} - r_k \cdot \sin(\gamma_k) \cdot \cos(\theta_{m,i}) - r_k \cdot \cos(\gamma_k) \cdot \sin(\theta_{m,i})$$
$$y_{v,k} = y_{m,i} + x'_{v,k} \cdot \sin(\theta_{m,i}) + y'_{v,k} \cdot \cos(\theta_{m,i}) = y_{m,i} + r_k \cdot \sin(\gamma_k) \cdot \sin(\theta_{m,i}) + r_k \cdot \cos(\gamma_k) \cdot \cos(\theta_{m,i})$$
$$\theta_{v,k} = \theta_{m,i} + \theta'_{v,k} = \theta_{m,i} + \gamma_k$$

<4.31>

Hay que tener en cuenta que en el anterior sistema de ecuaciones, el vector $P_{m,i} = \begin{bmatrix} x_{m,i} & y_{m,i} & \theta_{m,i} \end{bmatrix}^T$ es conocido, por lo que realmente sólo hay dos variables: r_k y γ_k.

El problema es que la expresión anterior (<4.31>) no tiene el formato típico de modelado del vector de salida presentado desde el principio del estudio ($\vec{z}_{v,k} = h\left(\vec{x}_{v,k}, \vec{v}_k\right)$), pues éste no muestra ninguna relación con el vector de estado ($\vec{x}_{v,k}$). El hecho es que, tal y como se planteó en los comentarios iniciales de este apartado, no se desea obtener un modelo específico del sistema de visión, sino que la salida $\vec{z}'_{v,k}$ del algoritmo SPL será preprocesada antes de ser incorporada al EKF con el fin de obtener $\vec{z}_{v,k}$.

Sin embargo, el algoritmo EKF sí requiere de una matriz de covarianza R que identifique al error de sensado del vector de salida ($\vec{z}_{v,k}$), y si la matriz de covarianza no está relacionada de forma lineal con este vector de salida o, como ocurre en esta aplicación, ni tan siquiera está relacionada con dicho vector de salida, será necesario buscar la matriz V_k que relacione de forma lineal el vector de ruido $\vec{v}_k'' = \begin{bmatrix} v_r & v_{\sin\gamma} & v_{\sin\gamma} \end{bmatrix}^T$ con el vector de salida $\vec{z}_{v,k}$.

El objetivo es idéntico al presentado a la hora de obtener el modelo del ruido de odometría: poder obtener la ecuación de salida (que procede en este caso del modelo de visión) de forma lineal, para poder aplicar así la definición del EKF tal y como se presentaba en la ecuación <3.14> y que aquí se repite:

$$h(\vec{x}_k) = h(\vec{x}_0) + C \cdot \left(\vec{x}_k - \vec{x}_0\right) + V_k \cdot \vec{v}_k = \vec{z}_k + V_k \cdot \vec{v}_k \qquad \text{<4.32>}$$

En dicha ecuación se ha eliminado la relación de modelado entre el vector de salida y el de estado (que no se requiere, tal y como ya se ha comentado), teniendo en cuenta que $\vec{x}_{v,k} = \vec{z}_{v,k}$, y que por tanto C es una matriz identidad. Además, la matriz de relación entre el vector de salida y el de ruido se calcula según se mostró ya en la ecuación <3.18>: $V_k \Rightarrow V_{[i,j]} = \dfrac{\partial h_{[i]}}{\partial v_{[j]}}\left(\vec{x}_k, 0\right)$. Finalmente, el vector de ruido de medida de la salida de posición será en este caso el ya mencionado, de componentes $\vec{v}_k'' = \begin{bmatrix} v_r & v_{\sin\gamma} & v_{\sin\gamma} \end{bmatrix}^T$.

Con todo ello se rescribe la ecuación <4.28>, a partir de la expresión <4.31> simplificada del siguiente modo:

$$\begin{aligned} x_{vn,k} &= cte1 - (r_k + v_r)\cdot(\sin(\gamma_k) + v_{\sin\gamma})\cdot cte2 - (r_k + v_r)\cdot(\cos(\gamma_k) + v_{\cos\gamma})\cdot cte3 \\ y_{vn,k} &= cte4 + (r_k + v_r)\cdot(\sin(\gamma_k) + v_{\sin\gamma})\cdot cte3 + (r_k + v_r)\cdot(\cos(\gamma_k) + v_{\cos\gamma})\cdot cte2 \qquad \text{<4.33>} \\ \theta_{vn,k} &= cte5 + \gamma_k \end{aligned}$$

Donde $P_{m,i} = \begin{bmatrix} x_{m,i} & y_{m,i} & \theta_{m,i} \end{bmatrix}^T = \begin{bmatrix} cte1 & cte4 & cte5 \end{bmatrix}^T$, $cte2 = \cos(\theta_{m,i})$ y $cte3 = \sin(\theta_{m,i})$.

Aplicando entonces la definición de la matriz V_k para completar el modelo buscado, se obtiene lo siguiente:

$$V_k = \begin{bmatrix} \dfrac{\partial x_k}{\partial v_r}\left(\vec{x}_0\right) & \dfrac{\partial x_k}{\partial v_{\sin\gamma}}\left(\vec{x}_0\right) & \dfrac{\partial x_k}{\partial v_{\cos\gamma}}\left(\vec{x}_0\right) \\ \dfrac{\partial y_k}{\partial v_r}\left(\vec{x}_0\right) & \dfrac{\partial y_k}{\partial v_{\sin\gamma}}\left(\vec{x}_0\right) & \dfrac{\partial y_k}{\partial v_{\cos\gamma}}\left(\vec{x}_0\right) \\ \dfrac{\partial \theta_k}{\partial v_r}\left(\vec{x}_0\right) & \dfrac{\partial \theta_k}{\partial v_{\sin\gamma}}\left(\vec{x}_0\right) & \dfrac{\partial \theta_k}{\partial v_{\cos\gamma}}\left(\vec{x}_0\right) \end{bmatrix} = \begin{bmatrix} -\sin(\gamma_k)\cdot cte2 & \sin(\gamma_k)\cdot cte3 & 0 \\ -r_k \cdot cte2 & r_k \cdot cte2 & 0 \\ -r_k \cdot cte3 & r_k \cdot cte2 & 0 \end{bmatrix} \qquad \text{<4.34>}$$

Una vez obtenida la matriz que relaciona $\vec{v}_k''$ con $\vec{z}_{v,k}$ solamente falta presentar la matriz de covarianza del ruido de salida R, que, tal y como ya se ha comentado se define perfectamente en la tesis doctoral que sirve de base de este trabajo.

En el trabajo desarrollado en la tesis se ha asegurado que la variable de ruido de medida $\vec{v}_k''$ asociado al vector de salida, tiene media nula, ya que cada vez que se arranca el proceso de visión se realiza una calibración de la cámara, tal y como comenta el autor ([Bouguet-99]). Esta previsión es equivalente al trabajo de corrección incorporado en el algoritmo de dead-reckoning que se presenta en puntos anteriores, y su efecto se observa claramente en la figura 4.13, donde se muestran las elipses de error del algoritmo de posicionamiento para el banco de imágenes ya comentado. En esta figura se observa claramente que todas las elipses de medida con error están centradas en el punto de prueba correspondiente.

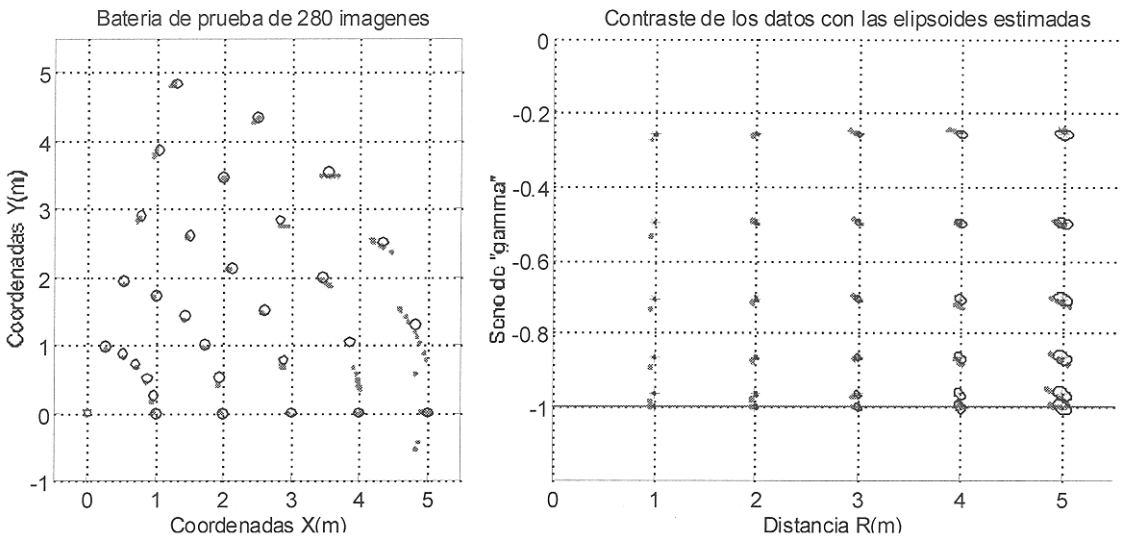


Figura 4.13. Resultado del ruido de medidas del sistema de visión con la batería de imágenes de prueba

La matriz de covarianza de ruido de salida (R_k) obtenida con el sistema de visión, tendrá por tanto la siguiente forma:

$$R_k = E\left[\vec{v}_k \cdot \vec{v}_k^T\right] = \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_{\sin\gamma}^2 & 0 \\ 0 & 0 & \sigma_{\cos\gamma}^2 \end{bmatrix} \qquad <4.35>$$

En la tesis [Gracía-01] se presenta el modo de calcular el valor instantáneo de esta matriz, que no se ha incluido en este documento pues incluye un proceso complejo de cálculo matricial que debe realizarse paso a paso y en función de muchas de las variables que entran en juego en el algoritmo SPL: el vector de orientación de la cámara con respecto a la marca ${}_{\lambda}\overleftarrow{\Omega}$, la posición de los centroides dentro de la imagen e incluso la distancia focal de la cámara, y evidentemente la covarianza del error de pixel. Va a ser por tanto,

y sin duda, una matriz de covarianza dinámica, tal y como se mostraba ya en su notación de la ecuación 4.40.

Así se puede completar el modelo del ruido de salida, es decir de la medida de posición obtenido con el sistema de visión, con la siguiente expresión:

$$h(\vec{x}_k) = h(\vec{x}_0) + C(\vec{x}_k - \vec{x}_0) + \ldots + V_k \cdot \vec{v}_k \text{ , con } V_k \cdot \vec{v}_k = \begin{bmatrix} -\sin(\gamma_k)\cdot cte2 & \sin(\gamma_k)\cdot cte3 & 0 \\ -r_k \cdot cte2 & r_k \cdot cte2 & 0 \\ -r_k \cdot cte3 & r_k \cdot cte2 & 0 \end{bmatrix} \cdot \begin{bmatrix} v_r \\ v_{\sin\gamma} \\ v_{\cos\gamma} \end{bmatrix}$$

<4.36>

estando el vector $\vec{v}_k$ definido por sus estadísticos de media nula y covarianza:

$$R_k = \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_{\sin\gamma}^2 & 0 \\ 0 & 0 & \sigma_{\cos\gamma}^2 \end{bmatrix}$$

5. Especificación de las técnicas de fusión EKF al modelo de silla de ruedas automatizada

LLegados a este punto, es necesario realizar una recapitulación de lo presentado en el documento en los capítulos anteriores

- Ante el objetivo de partida de este trabajo de investigación, se muestran en el primer capítulo distintas alternativas de fusión de datos, concluyendo que la más adecuada para la aplicación que interesa (posicionamiento de robots móviles), es el estimador Bayesiano, óptimo y recursivo denominado Filtro de Kalman (KF).

- Dicho planteamiento da lugar en el capítulo siguiente al desarrollo y justificación matemática del algoritmo de fusión, que resulta necesitar de dos etapas de procesamiento: una de estimación y una de posterior corrección. Se observa además, cómo esta organización da pie a la aplicación de fusión de filtro: en la etapa de predicción se empleará uno de los sensores (concretamente el odométrico) para obtener la estimación de posición, y en la de corrección se completará la estimación de posición con la información proporcionada por el algoritmo de visión. Además, en previsión de que los sistemas a fusionar sean no lineales se realiza también el estudio de la versión del KF para este tipo de sistemas, lo que lleva al desarrollo del Filtro de Kalman Extendido (EKF).

- Finalmente, en siguiente capítulo se analiza en detalle el sistema que se desea fusionar, con el fin de obtener el modelo matemático de los dos sistemas sensoriales que han de ser incorporados en el algoritmo de fusión. Estos modelos son muy particulares, pues por un lado tienen que incluir información sobre el ruido de medidas de ambos sensores y por otro han de presentar un formato lineal para poder ser introducidos en la algoritmia de predicción y corrección planteada por el filtro.

Con toda esta información, en el presente capítulo se aborda la tarea de relacionar el algoritmo de Kalman con el modelo del sistema, y de llevar a cabo, a modo de ejemplo diferentes simulaciones sobre la

mencionada fusión. Para poder implementar dicha simulación se ha utilizado el programa MATLAB, y para poder desarrollar la evolución del filtro en tiempo pseudo-real se ha tenido que implementar el algoritmo utilizando su herramienta Real Time Workshop (RTW).

En los siguientes apartados se abordarán distintas simulaciones del planteamiento propuesto que permitirán sacar conclusiones para la posterior implementación del algoritmo de filtrado en un sistema empotrado a bordo del robot, que recoja realmente las señales sensoriales a fusionar.

5.1. Aplicación de los modelos obtenidos al algoritmo de fusión EKF

El desarrollo realizado en el capítulo 3 tiene como resultado la algoritmia que permite implementar el EKF. Para poder aplicarlo al sistema de interés es necesario incorporarle las matrices de los modelos de posicionamiento (odométrico y de visión con sus correspondientes señales de ruido). Para llevar a cabo este paso se va a descomponer la matemática del EKF en sus dos partes fundamentales:

5.1.1. La etapa de predicción del EKF (k)

El primer paso a realizar en esta etapa consiste en calcular el vector de estados del sistema, que coincide, tal y como ya se ha visto, con la salida de posición del robot $\vec{x}_k = \vec{z}_k = \begin{bmatrix} x_k & y_k & \theta_k \end{bmatrix}^T$. La estimación de esta señal de posición se hace con el modelo discretizado y no lineal de uno de los sistemas sensoriales, concretamente el sistema odométrico ($\hat{x}_{o,k+1/k}$), que además es el único del que se ha obtenido un modelo matemático completo. De este modo, se llega a la siguiente asociación entre el algoritmo EKF y los modelos obtenidos:

$$\hat{x}_{k+1/k} = f(\hat{x}_k, \vec{u}_k, 0) \Longrightarrow \begin{aligned} \hat{x}_{o,k+1/k} &= \hat{x}_{o,k} + T_s \cdot V_k \cdot \cos\theta_k \\ \hat{y}_{o,k+1/k} &= \hat{y}_{o,k} + T_s \cdot V_k \cdot \sin\theta_k \\ \hat{\theta}_{o,k+1/k} &= \hat{\theta}_{o,k} + T_s \cdot \Omega_k \end{aligned} \qquad <5.1>$$

Donde V_k y Ω_k se obtienen en cada instante directamente del vector de sensado odométrico $\vec{\varpi} = \begin{bmatrix} \omega_D & \omega_I \end{bmatrix}^T$, a través de las ecuaciones de cinemática inversa:

$$\begin{aligned} V_k &= \frac{R}{2}\left(\omega_D + \omega_I\right) \\ \Omega_k &= \frac{R}{D}\left(\omega_D - \omega_I\right) \end{aligned} \qquad <5.2>$$

Por otro lado, hay que destacar que el vector $\hat{x}_k$, representa a la salida global del estimador, es decir, al vector de estados corregido en la etapa de corrección del periodo anterior del algoritmo recursivo (lo que también se expresa como $\hat{x}_{k/k}$).

Es interesante observar el hecho de que se ha presupuesto que estas medidas no tienen ruido, ya que no este no se incluye junto al modelo de estados del sistema.

La información acerca del ruido que incorporan estas señales sensadas se introduce en el EKF en el siguiente paso de la etapa predicción, que consiste en obtener la parte de innovación de la matriz de covarianza del error de estimación:

$$P_{k+1/k} = A_k \cdot P_{k/k} \cdot A_k^T + W_k \cdot Q \cdot W_k^T \quad <5.3>$$

Donde:

- A_k es la matriz de estados del modelo de estimación (el odométrico) linealizado (ha de ser de tamaño 3x3), cuyo valor instantáneo desarrollado en el capítulo anterior es el siguiente:

$$A_k = \begin{bmatrix} 1 & 0 & -V_k \cdot T_s \cdot \sin(\theta_k) \\ 0 & 1 & V_k \cdot T_s \cdot \cos(\theta_k) \\ 0 & 0 & 1 \end{bmatrix} \quad <5.4>$$

Merece la pena destacar en la matriz anterior, que se han sustituido los subíndices de "punto de trabajo" (m_o) que se usaron en las expresiones del modelado lineal del capítulo anterior, por la notación `^' indicadora de que se trata de valores estimados. Esta sustitución (que será aplicada en todos los casos de linealización en este capítulo), hará más precisos los resultados de linealización.

- $P_{k/k}$ es la matriz de covarianza del error de estimación obtenida en la etapa de corrección del mismo instante k en el que se desarrolla la estimación para el instante k+1 (ha de ser por tanto de tamaño 3x3). En caso de ser el primer periodo de ejecución $P_{0/0}$ se inicializa a cero, ya que se considera conocido con toda certeza (incertidumbre de estimación nula) el valor del vector de estado, que es en este caso la posición inicial del robot. En las simulaciones se observará el efecto de asignar un valor inicial distinto a esta matriz.

- Y finalmente Q y W_k son, respectivamente la matriz de covarianza del error de estados, y la matriz que permite relacionar de forma lineal el vector de ruido de medida del estado con el propio vector de estado (todo ello referente a las medidas de ruido odométricas). En el análisis realizado en el capítulo anterior se plantean dos posibilidades para estas matrices:

1. Que el vector de ruido de estados sea $\vec{w}_k = \begin{bmatrix} w_x & w_y & w_\theta \end{bmatrix}^T$, en cuyo caso la matriz W_k es constante e igual a la matriz identidad, ya que el ruido de medidas se expresa directamente sobre los distintos elementos del vector de estado. Por otra parte, en este caso el valor de la matriz Q (que deberá ser en este caso de tamaño 3x3), se ha obtenido empíricamente tal y

como se comentaba en el capítulo anterior (eliminando previamente el valor medio del ruido), con una forma final como la que sigue:

$$Q = \begin{bmatrix} 0.000438 & 0 & 0 \\ 0 & 0.0014 & 0 \\ 0 & 0 & 0.00030625 \end{bmatrix} \qquad <5.5>$$

2. Que el vector de estado sea $\vec{w}_k = \begin{bmatrix} w_D & w_I \end{bmatrix}^T$, en cuyo caso W_k relaciona de forma lineal este vector con el de estados, a través del modelo del sistema odométrico y de las ecuaciones de cinemática inversa antes mostradas. El resultado es el que se muestra a continuación:

$$W_k = \begin{bmatrix} R/2 \cdot T_s \cdot \cos(\theta_k) & R/2 \cdot T_s \cdot \cos(\theta_k) \\ R/2 \cdot T_s \cdot \sin(\theta_k) & R/2 \cdot T_s \cdot \sin(\theta_k) \\ R/D \cdot T_s & -R/D \cdot T_s \end{bmatrix} \qquad <5.6>$$

El valor de la matriz Q (que en este caso ha de ser de 2x2) no se obtuvo en el desarrollo de la investigación, pero su cálculo se realizaría, sin demasiada dificultad, de forma empírica, de igual modo que se hizo con el otro modelo del ruido de estados, resultando en ese caso de la siguiente forma:

$$Q = \begin{bmatrix} 0.000438 & 0 & 0 \\ 0 & 0.0014 & 0 \\ 0 & 0 & 0.00030625 \end{bmatrix}$$

El hecho de elegir uno u otro modelo para el ruido de estados depende únicamente de cómo se haya representado su matriz de covarianza. Como en este trabajo se ha obtenido empíricamente el valor de la matriz de covarianza del ruido de estados directamente sobre el vector de posición o de estados, se debe utilizar el primer método para modelar el ruido. Sin embargo ambas posibilidades serán simuladas en las pruebas que se presenten a continuación, pudiendo de este modo comprobar el efecto de usar uno u otro modelo.

Lo que sí se puede adelantar desde este momento es que si se desea incorporar dinamismo al modelado del ruido se ha de usar el segundo modelo (en él W_k depende del estado o posición del robot) o hacer que la matriz de covarianza Q utilizada con el primer modelo sea recalculada en cada tramo del recorrido. Con este simple planteamiento se puede observar fácilmente que va a ser más efectivo utilizar el modelo segundo.

5.1.2. La etapa de corrección del EKF (k+1)

La etapa de corrección del algoritmo comienza con el cálculo de la matriz de estimación, la también llamada ganancia de Kalman, que responde a la siguiente expresión:

$$K_{k+1} = P_{k+1/k} \cdot C_{k+1}{}^{T} \left(C_{k+1} \cdot P_{k+1/k} \cdot C_{k+1}^{T} + V_{k+1} \cdot R \cdot V_{k+1}^{T} \right)^{-1} \qquad <5.7>$$

Donde:

- $P_{k+1/k}$ es la matriz de innovación de la covarianza del error de estimación que se obtuvo en la etapa de predicción (en el instante k) tal y como se presentó en el apartado anterior (ecuación <5.3>).

- C_{k+1} es la matriz del modelo de estados (modelo odométrico, en este caso) discreto y lineal que relaciona el vector de estados $\vec{x}_{o,k}$ con el vector de salida $\vec{z}_{o,k}$ (ha de ser de tamaño 3x3). Como en el caso de interés estos dos vectores coinciden, la matriz tiene por valor el de la identidad. Debido a esto, el valor de la ganancia de Kalman puede rescribirse de la siguiente forma (agilizando así la ejecución en tiempo real del algoritmo):

$$K_{k+1} = P_{k+1/k} \left(P_{k+1/k} + V_{k+1} \cdot R \cdot V_{k+1}^{T} \right)^{-1} \qquad <5.8>$$

- R y V_{k+1} son las matrices que caracterizan al ruido de salida, lo que significa en el sistema de interés de este trabajo, que caracterizan al ruido de posicionamiento del sistema de visión. Son, por tanto, equivalentes a las anteriormente descritas Q y W_{k+1}, respectivamente, pero en este caso referidas al otro sistema sensorial a fusionar. La obtención de las matrices se hace tal y como se mostró en el capítulo anterior, a través del desarrollo realizado en la tesis [García-01]. Concretamente la matriz V_{k+1} que relaciona de forma lineal el vector de salida $\vec{z}_{v,k}$ con el de ruido (que se define, tal y como ya se comentó, a través de las variables de posición transformadas expresadas en coordenadas polares $\vec{v}_k'' = \begin{bmatrix} v_r & v_{\sin\gamma} & v_{\sin\gamma} \end{bmatrix}^T$) tiene finalmente la siguiente forma (será de tamaño 3x3):

$$V_{k+1} = \begin{bmatrix} -\sin\gamma_{k+1} \cdot cte2 & \sin\gamma_{k+1} \cdot cte3 & 0 \\ -r_{k+1} \cdot cte2 & r_{k+1} \cdot cte2 & 0 \\ -r_{k+1} \cdot cte3 & r_{k+1} \cdot cte2 & 0 \end{bmatrix} \qquad <5.9>$$

Donde la posición absoluta de la marca, obtenida del mapa del entorno y utilizada para obtener información de posición del segundo sistema de sensado a fusionar (el de visión) proporciona el valor de las constantes aplicando las transformaciones mostradas en el capítulo anterior, de este modo: $P_{m,i} = \begin{bmatrix} x_{m,i} & y_{m,i} & \theta_{m,i} \end{bmatrix}^T = \begin{bmatrix} cte1 & cte4 & cte5 \end{bmatrix}^T$, $cte2 = \cos(\theta_{m,i})$ y $cte3 = \sin(\theta_{m,i})$.

En la matriz V_{k+1} aparecen, además, las variables de salida transformadas del algoritmo SPL (el vector en coordenadas polares $\vec{z}_{v,k+1}'' = \begin{bmatrix} r_{k+1} & \sin(\gamma_{k+1}) & \cos(\gamma_{k+1}) \end{bmatrix}^T$), lo que significa que para poder realizar la integración del error de medidas de posicionamiento por visión en el EKF, será necesario incorporar este vector. Este hecho se repite a la hora de obtener el valor de la matriz de covarianza del error de medidas R, ya que tal y como se presentó en el capítulo anterior, ésta depende fuertemente del mencionado vector de salida del algoritmo SPL (el vector de orientación de la cámara con respecto a la marca, la posición de los centroides dentro de la imagen e incluso la distancia focal de la cámara, y evidentemente la covarianza del error de pixel). El valor de esta última matriz tendrá, tal y como ya se

comentó anteriormente, la siguiente forma (ha de ser también de tamaño 3x3), recalcando el hecho de que se trata de una matriz dinámica:

$$R_k = \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_{\sin\gamma}^2 & 0 \\ 0 & 0 & \sigma_{\cos\gamma}^2 \end{bmatrix} \quad <5.10>$$

En la tesis [Gracía-01] se presenta el modo de calcular el valor instantáneo de la matriz R_{k+1} a partir de los parámetros y variables comentados, del sistema de visión.

A la vista de las expresiones de R_{k+1} y V_{k+1} se observa claramente que ambas son dinámicas, es decir, dependen de la posición instantánea del robot. Este hecho que puede ralentizar el procesamiento del filtrado de Kalman es, por otra parte, muy ventajoso, pues del mismo modo que se comentaba al presentar el segundo modelo del ruido de odometría, este modelo caracterizará más fielmente que un modelo estático a la variable de error que hace evolucionar correctamente al estimador.

El siguiente paso en el desarrollo de la etapa de corrección es la corrección propiamente dicha del vector de estados estimados, con la siguiente expresión recuperada de capítulo 3:

$$\hat{x}_{k+1/k+1} = \hat{x}_{k+1/k} + K_{k+1}\left(\vec{z}_{k+1} - h(\hat{x}_{k+1/k},0)\right) \quad <5.11>$$

Donde:

- $\hat{x}_{k+1/k}$ es, realmente, la estimación del vector de estados obtenida con el modelo odométrico (discretizado y no lineal) en la etapa de predicción del algoritmo EKF ($\hat{x}_{o,k+1/k}$) (ecuación <5.1>).

- K_{k+1} la ganancia de Kalman presentada anteriormente (ecuación <5.8>).

- z_{k+1} es la salida de posición obtenida a través del sistema de visión, es decir, el vector $\vec{z}_{v,k+1} = \begin{bmatrix} x_{v,k+1} & y_{v,k+1} & \theta_{v,k+1} \end{bmatrix}^T$. Tal y como se comentaba en el capítulo anterior, esta información se obtiene aplicando el siguiente sistema de transformación al vector de salida directo del algoritmo SPL $\vec{z}'_{v,k+1} = \begin{bmatrix} x'_{v,k+1} & y'_{v,k+1} & \gamma_{k+1} \end{bmatrix}^T$:

$$\begin{aligned} x_{v,k+1} &= x_{m,i} + x'_{v,k+1} \cdot \cos(\theta_{m,i}) - y'_{v,k+1} \cdot \sin(\theta_{m,i}) \\ y_{v,k+1} &= y_{m,i} + x'_{v,k+1} \cdot \sin(\theta_{m,i}) + y'_{v,k+1} \cdot \cos(\theta_{m,i}) \\ \theta_{v,k+1} &= \theta_{m,i} + \gamma_{k+1} \end{aligned} \quad <5.12>$$

Donde aparecen de nuevo las coordenadas absolutas de posición $P_{m,i} = \begin{bmatrix} x_{m,i} & y_{m,i} & \theta_{m,i} \end{bmatrix}^T$ de la marca 'i' utilizada por el algoritmo SPL para posicionar el móvil.

Es interesante recordar (del capitulo 3) el hecho de que es justo en este punto donde se produce la fusión de las medidas de posición obtenidas por los dos sistemas modelados: Las odométricas se utilizaron ya en la etapa de predicción; y en el cálculo del vector de estados estimado y corregido, se añaden las medidas de posición del sistema de visión. Además, tal y como se muestra en la ecuación <5.11> de corrección del vector de estados, la fusión se realiza comparando esta medida de visión ($\vec{z}_{v,k+1}$) con la estimación de la salida proporcionada por las medidas odométricas ($\vec{z}_{o,k+1}$), constituyendo lo que se conoce como "vector residual" y que es ponderado por la ganancia de Kalman para incorporarse a la predicción realizada en la etapa anterior.

Por su parte, el ruido asociado a los dos sistemas de posicionamiento también se fusiona: el de las medidas odométricas en la etapa de predicción (tal y como se mostró en el cálculo de la matriz de innovación $P_{k+1/k}$, ecuación <5.3>), y el de las medidas de visión en el cálculo de la ganancia de Kalman en esta etapa de corrección (ecuación <5.8>).

- Finalmente $h(\hat{x}_{k+1/k},0)$ se corresponde con el vector de salida del modelo odométrico ($\hat{z}_{o,k+1}$ que no tiene en cuenta el ruido, como ya se comentó varias veces en el documento) y que, teniendo en cuenta lo presentado anteriormente sobre la ecuación de salida odométrica, y más concretamente sobre la matriz C_{k+1}, coincide con el vector de estados de este sistema $\hat{x}_{o,k+1/k}$. Su cálculo se realiza, por tanto, a partir de las ecuaciones discretizadas y no lineales del dead-reckoning presentadas en este capítulo, en el primer paso de la etapa de predicción (ecuaciones <5.1>), y sirve para obtener el mencionado "vector de residuos".

El último paso a desarrollar en la etapa de predicción y que daría lugar a la conclusión de una vuelta en la evolución recursiva del algoritmo de fusión, es la actualización de la matriz de covarianza del error de estimación, mediante la siguiente expresión:

$$P_{k+1/k+1} = P_{k+1/k} - K_{k+1} \cdot C_{k+1} \cdot P_{k+1/k} \qquad <5.13>$$

En la que todas las matrices a las que implica han sido ya presentadas en este apartado o en el anterior. Simplemente merece la pena destacar el hecho de que al ser la matriz C_{k+1} de valor igual a la matriz identidad, la anterior expresión puede simplificarse del siguiente modo:

$$P_{k+1/k+1} = P_{k+1/k} - K_{k+1} \cdot P_{k+1/k} \qquad <5.14>$$

5.2. Implementación en tiempo pseudo-real del algoritmo de fusión

En el apartado anterior, se establece el desarrollo final del EKF asociando cada una de las variables del estimador con los modelos de los sistemas sensoriales correspondientes. Sin embargo quedan un par de cuestiones por resolver a la hora de llevar a la práctica la implementación del estimador óptimo.

Por un lado es necesario comentar el hecho de que los dos sistemas sensoriales no se desarrollan con el mismo periodo de repetición, aunque hasta el momento no se haya realizado ninguna distinción

relativa a este aspecto. Para tener en cuenta esta diferencia se va a nombrar de distinto modo al periodo de muestreo (o de ejecución) de los dos sistemas, de este modo:

- T_{so} , será el periodo de muestreo del sistema odométrico.
- T_{sv} , será el periodo de muestreo del sistema de visión, tal que $T_{sv} = n \cdot T_{so}$ (son múltiplos).

El periodo de ejecución del filtro será el del sistema más rápido, es decir coincidirá con el del modelo odométrico. La diferencia de tiempos de ejecución, en la implementación real del EKF, se verá reflejada solamente en la etapa de corrección del algoritmo EKF, pues en la de predicción sólo entra en juego el modelo odométrico que evolucionará a la misma frecuencia que el algoritmo de fusión.

Sin embargo, en la etapa de corrección aparecerá una diferencia, y es que aunque ésta se ejecute con un periodo de muestro de valor T_{so} , solamente incorporará nuevas medidas del sistema de visión cada 'n' veces (siendo 'n' el número de veces que T_{sv} es múltiplo de T_{so}). Así los tres elementos (variables y matrices) que caracterizan al sistema de visión permanecerán constantes durante los $n \cdot T_{so}$ segundos que dure su periodo de ejecución. En otras palabras, el algoritmo de fusión dará una salida solamente cada T_{sv} , aunque hará evolucionar el vector de estados odométrico de la etapa de corrección (que seguirá ejecutándose en todos los periodos de ejecución del algoritmo), cada T_{so} .

En la simulación del algoritmo de fusión se observan las consecuencias de este factor, que no serán otras que la ralentización e incluso la divergencia en algunos casos del algoritmo de Kalman. Además, debido a este hecho el ruido del modelo de odometría será especialmente importante, pues tendrá efecto durante más tiempo en el estimador, llegando incluso a desestabilizar la convergencia de la ganancia de Kalman. En el apartado siguiente se verá con diferentes ejemplos éste aspecto.

Con todo ello, las ecuaciones finales que han sido programadas para la implementación final y simulación del algoritmo son las que se muestran a continuación (figura 5.1):

En dicha figura se ha mostrado mediante los subíndices de las variables correspondientes, no sólo el origen odométrico o de visión de la información de posición de la variable (o del resultado de la fusión, que se indica a partir de ahora con el subíndice 'f'), sino también el periodo de muestreo utilizado para actualizar cada una de las mismas ('kv' si se actualiza cada T_{sv} y 'ko' si se hace cada T_{so}). Desde este punto de vista merece la pena destacar la nomenclatura utilizada para el vector de estados corregido ($\hat{x}_{k/k}$) necesario para calcular la estimación del mismo ($\hat{x}_{o,ko+1/ko}$) en la etapa de predicción. Tal y como se observa en la mencionada figura, se han incluido dos subíndices ('v,kv' o 'o,ko'), indicando que se aplicará uno u otro en función de si se posee información actualizada en dicho periodo sobre el vector corregido (utilizando entonces $\hat{x}_{v,kv}$ como $\hat{x}_{k/k}$) o no, en cuyo último caso se utilizará como vector de estados corregido el predicho en el anterior periodo de repetición odométrico ($\hat{x}_{o,ko/ko-1}$).

Además, también se ha incluido en la figura 4.1 anterior, información acerca del instante en que el hilo de ejecución pasa de una a otra etapa del EKF, mostrándose cómo mientras no se produzca la

repetición del periodo de muestreo del sistema de visión T_{sv}, se seguirá ejecutando solamente la etapa de predicción del algoritmo.

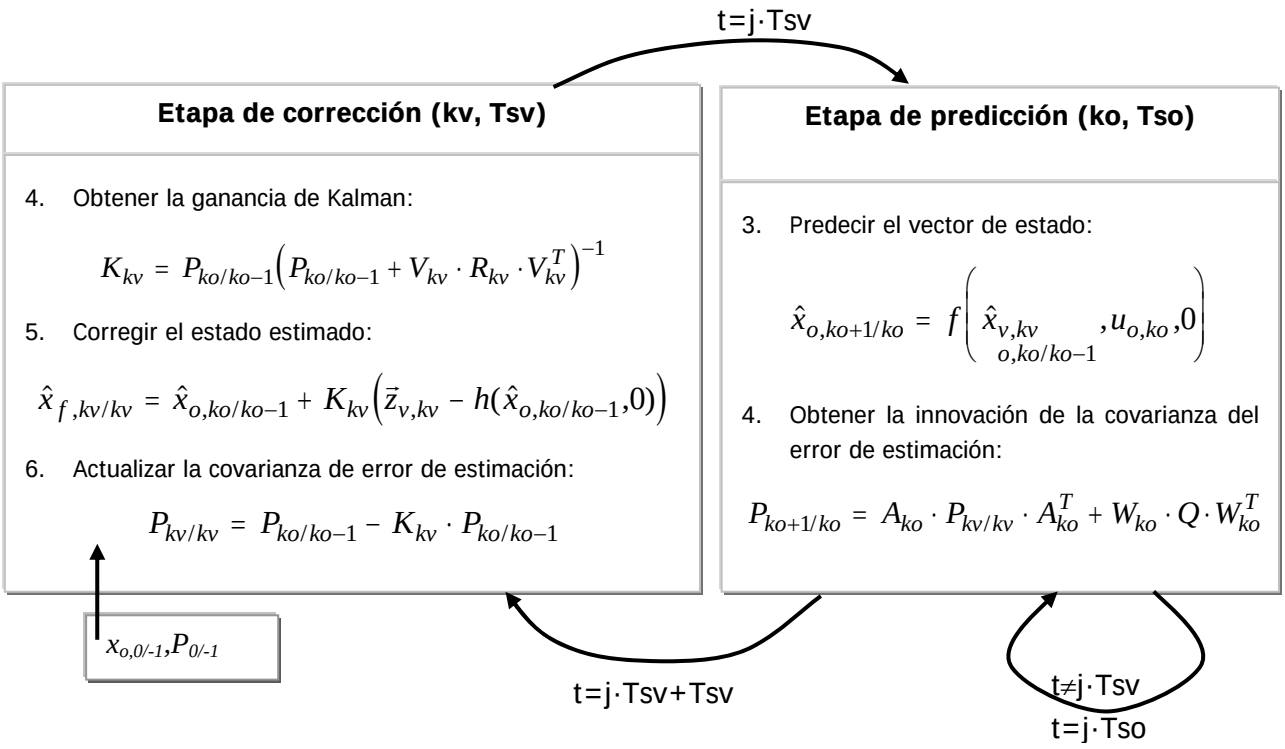


Figura 5.1. Diagrama explicativo de la implementación final del algoritmo de fusión para posicionamiento de robots móviles basado en el EKF.

Además se ha incluido también ya, en la misma figura 5.1, la modificación de la secuencialidad en la ejecución de las dos etapas del algoritmo de fusión, que es el otro aspecto importante que merece la pena destacar de la implementación real del EKF. Tal y como se puede imaginar, en cada periodo de ejecución primero se desarrollará la etapa de corrección y luego la de predicción, que prepara el algoritmo para el siguiente periodo del estimador recursivo. Es decir, la implementación real del EKF se hace en el orden inverso a como se ha desarrollado en este documento. Este hecho obliga a tener una etapa de inicialización del algoritmo que consiste fundamentalmente en establecer el valor inicial predicho del vector de estado (del modelo odométrico $\hat{x}_{o,0/-1}$) a través del conocimiento de la posición de partida del robot en su tarea de navegación, y de la matriz $P_{0/-1}$ de la forma que ya se ha comentado en apartados anteriores.

5.3. Resultados obtenidos

Con todo lo comentado, se aborda la programación del algoritmo de fusión resultante, y su posterior incorporación a una herramienta de simulación que permita obtener conclusiones acerca del mismo, antes de llevarlo a la práctica mediante su exportación a la plataforma final de navegación.

Tal y como ya se ha comentado el algoritmo ha sido desarrollado en ‘C’ para Real Time Workshop de MATLAB, de modo que se puede incorporar como un bloque más al modelado del sistema real. La figura 5.2 muestra la apariencia final del bloque de Simulink cerrado y abierto, observándose de este último modo los parámetros de configuración que pueden incluirse en el mismo.

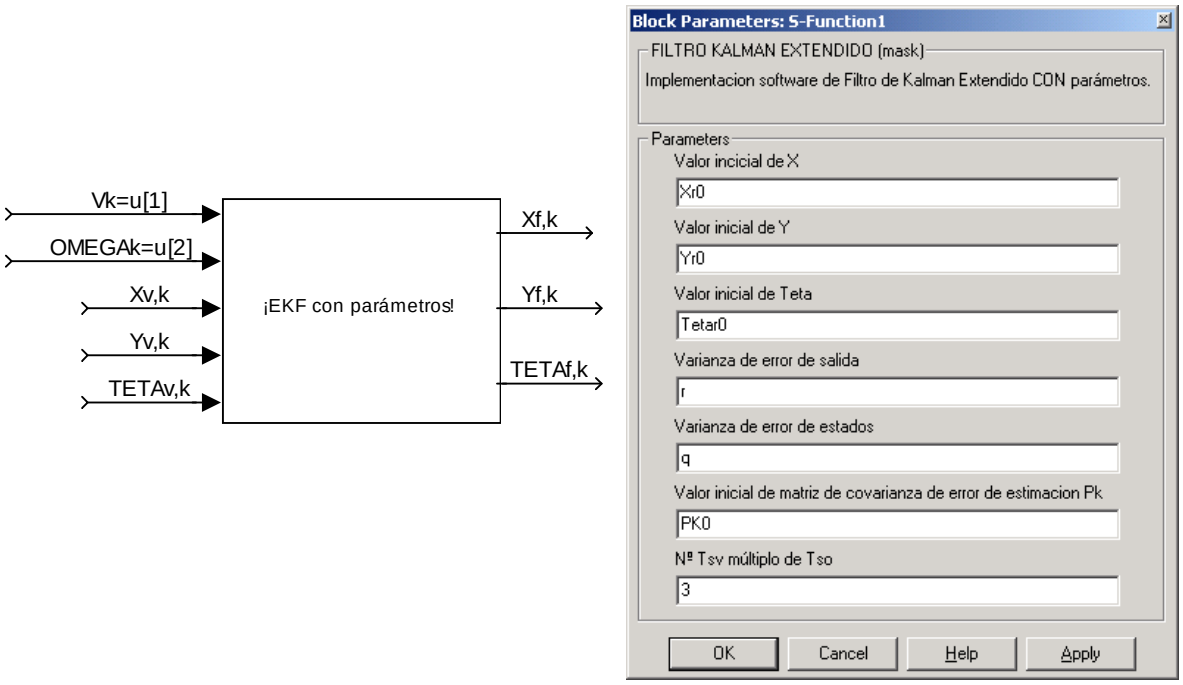


Figura 5.2. Apariencia final del algoritmo de fusión implementado

Merece la pena realizar un pequeño comentario sobre los parámetros de simulación, sobre todo del hecho de que se han incluido, además de los valores de inicialización del algoritmo $\hat{x}_{o,0/-1}$ y $P_{0/-1}$, y del valor de la matrices de la covarianza de ruido (supuesta constante) de medidas de estado (odométrico) y de salida (visión), el parámetro que indica la multiplicidad de T_{sv} con respecto a T_{so}.

Ha de quedar claro además, que el algoritmo de fusión ha sido desarrollado específicamente para el sistema global cuyo modelo se ha presentado en este documento. Sin embargo, en caso de querer generalizar su uso para cualquier otro sistema la portabilidad de la programación conseguida en este trabajo facilitará su exportación en todos los casos.

Además, para llevar a cabo la simulación del algoritmo se ha desarrollado en Simulink un ejemplo de modelado del que podría ser el algoritmo de generación de trayectorias del móvil, para el que se implementa la fusión y estimación de la posición. En el apartado siguiente se muestra el modelado del mismo, y después de ello, las distintas simulaciones realizadas con el fin de obtener conclusiones a cerca del trabajo desarrollado.

5.3.1. Modelado del generador y controlador de trayectorias

Finalizado el estudio y desarrollo global del algoritmo de fusión para el modelo de interés, pareció interesante incluir en el trabajo una simulación de su comportamiento en el uso específico que se le pretendía dar: la estimación de la posición del robot móvil en apoyo al generador y controlador de trayectorias del sistema de navegación global.

Con este objetivo, se desarrolló un modelo del generador y controlador de trayectorias, que históricamente iba a estar gobernado por la realimentación de posición que proporcionaba el sistema sensorial odométrico [Marrón–00]. Para no hacer el estudio muy extenso, concretamente se desarrolló el modelo de un generador de rectas, como trayectoria básica.

Dicho modelo, cuya representación en diagrama de bloques se incluye en la figura 5.4, muestra varias partes bien diferenciadas:

- **Generador de trayectorias o consigna,** que permite obtener la consigna de posición instantánea, necesaria para generar la señal de error sobre la que se soporta el controlador, a partir de la posición real del robot (inicialmente obtenida sólo por odometría) y del punto inicial y final del la trayectoria recta. El desarrollo de este bloque se hizo en base a los planteamientos expuestos en [Marrón-00].

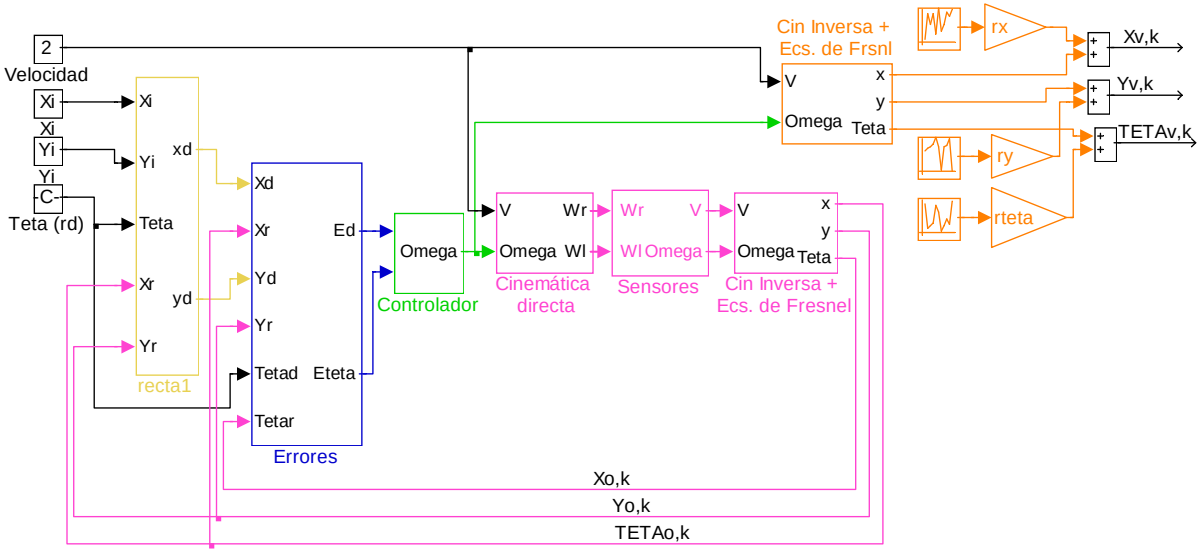


Figura 5.4. Modelo del generador y controlador de trayectorias diseñado para las pruebas del EKF

- **Elemento de obtención del vector de error,** que realiza su función mediante la comparación del vector de consigna de posición instantánea con el de posición real odométrica.

- **Controlador de posición,** que permite obtener el valor de consigna de velocidad de giro para el bajo nivel mediante un algoritmo clásico de control (PI) aplicado a la señal de error. Merece la pena destacar el hecho de que la velocidad lineal (V) en el modelo se ha supuesto constante en el

desarrollo de toda la trayectoria, pues es una técnica muy extendida en este tipo de controladores (en algunos casos se modifica pero generalmente no de forma lineal sino a tramos y muy puntualmente, de modo que no se considera una variable del sistema [Marrón-00]). Este hecho coincide con lo supuesto en el capítulo de modelado, a la hora de realizar la linealización, lo cual valida la suposición hecha en su momento.

- **Modelo de la planta,** que se realiza a través de la odometría, tal y como ya se ha comentado repetidas veces en el documento. Merece la pena destacar en este punto, el modo en que se han modelado los dos sistemas sensoriales de posición. Si bien ambos parten del modelo de dead-reckoning, el ruido (que es realmente el que caracteriza a uno u otro modelo de sensado) es independiente para cada uno de ellos, generándose así las dos salidas de posición independientes. En las simulaciones, el ruido asociado al sistema de visión se ha incorporado directamente sobre el vector de posición (ver anterior figura 5.4), mientras que para el ruido de odometría se han generado dos modelos, uno que lo incorpora también sobre el propio vector de salida, y otro que lo incluye en las variables de velocidad angular (ver figura 5.5), a fin de observar las dos posibilidades contempladas en anteriores capítulos.

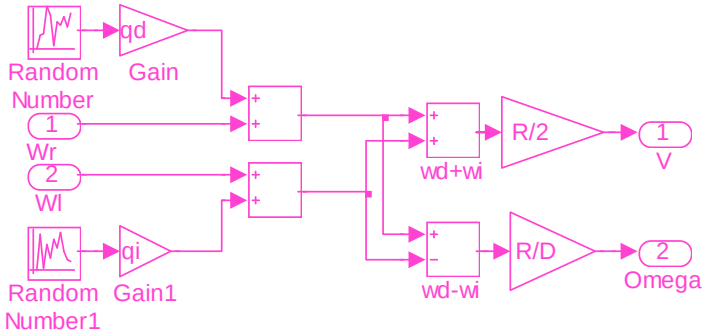


Figura 5.5. Modelo del ruido odométrico sobre las velocidades angulares del robot

A la hora de obtener las variables de error se ha propuesto un nuevo modelo, que es utilizado en muchas ocasiones con el objetivo de simplificar el sistema de posición (ésta se hace local) también para desarrollar el modelo de posicionamiento.

El modelo propuesto mantiene el vector de entrada y de salida, pero se toma como vector de estado el error de orientación y de desplazamiento del móvil ($\vec{x} = \begin{bmatrix} d_e & \theta_e \end{bmatrix}^T$ [9]) con respecto al punto deseado de una trayectoria a recorrer ($P_d = \begin{bmatrix} x_d & y_d & \theta_d \end{bmatrix}^T$). En la figura 5.6 se muestra el modo de obtener la expresión de este nuevo vector, que se presenta a continuación.

[9] En todas las expresiones que se incluyen a continuación para el desarrollo del nuevo modelo odométrico, se omiten los subíndices 'o' y 'k', teniendo en cuenta que como el modelo no se va a usar en el desarrollo del EKF, este hecho no va a oscurecer las ecuaciones.

$$d_e = -(x_o - x_d)\cdot\sin(\theta_d) + (y_o - y_d)\cdot\cos(\theta_d)$$
$$\theta_e = \theta_o - \theta_d \qquad <5.15>$$

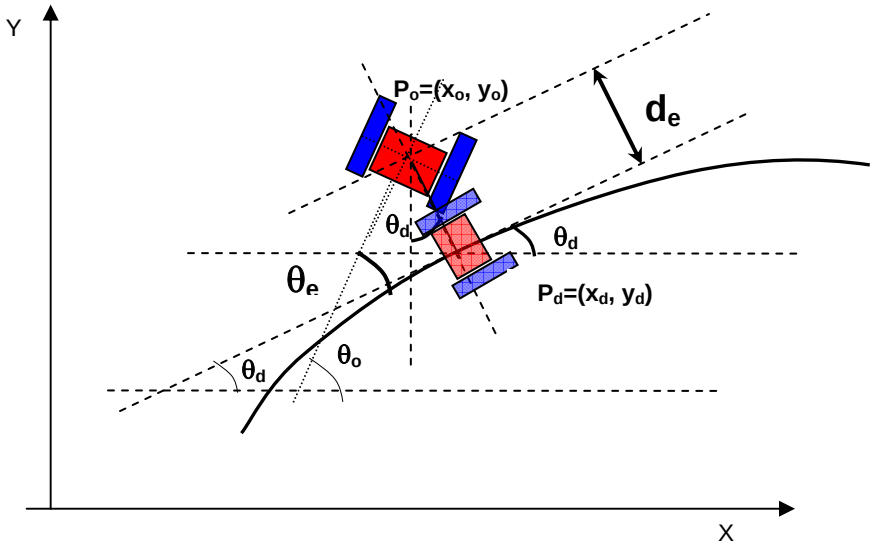


Figura 5.6. Diagrama explicativo del nuevo modelo del robot móvil

De este modo el modelo de la planta vendría, en este caso, descrito por las siguiente ecuaciones de estado y de salida:

$$\begin{bmatrix} \dot{d}_e(t) \\ \dot{\theta}_e(t) \end{bmatrix} = A\cdot\begin{bmatrix} d_e(t) \\ \theta_e(t) \end{bmatrix} + B\cdot\begin{bmatrix} V(t) \\ \Omega(t) \end{bmatrix}$$
$$\begin{bmatrix} x_o(t) \\ y_o(t) \\ \theta_o(t) \end{bmatrix} = C\cdot\begin{bmatrix} d_e(t) \\ \theta_e(t) \end{bmatrix} + D\cdot\begin{bmatrix} V(t) \\ \Omega(t) \end{bmatrix} \qquad <5.16>$$

Donde $D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$ y el vector de estados derivados $\dot{\vec{x}} = \begin{bmatrix} \dot{d}_e & \dot{\theta}_e \end{bmatrix}^T$. Teniendo en cuenta que a efectos de derivación temporal la posición deseada $P_d = \begin{bmatrix} x_d & y_d & \theta_d \end{bmatrix}^T$ (consigna del generador de trayectorias) es constante en cada intervalo de ejecución, el valor del vector $\dot{\vec{x}}$ es el siguiente:

$$\begin{aligned}\dot{d}_e &= -\dot{x}_o \cdot \sin(\theta_d) + \dot{y}_o \cdot \cos(\theta_d) \\ &= -V\cos(\theta_o)\cdot\sin(\theta_d) + V\sin(\theta_o)\cdot\cos(\theta_d) \\ &= V\big(\sin(\theta_o)\cdot\cos(\theta_d) - \cos(\theta_o)\cdot\sin(\theta_d)\big) \\ &= V\sin(\theta_o - \theta_d) = V\sin(\theta_e) \\ \dot{\theta}_e &= \dot{\theta}_o = \Omega\end{aligned} \qquad <5.17>$$

Este nuevo modelo odométrico del móvil tampoco es lineal, ya que, según lo expuesto en la anterior ecuación, sus variables de estado se describen tal y como se presenta a continuación:

$$\begin{aligned}\dot{d}_e(t) &= V(t)\cdot\sin\theta_e(t) \\ \dot{\theta}_e(t) &= \Omega(t)\end{aligned} \qquad <5.18>$$

Y su expresión en el espacio discreto sería en este caso el siguiente:

$$\begin{aligned}d_{e,k} &= d_{e,k-1} + T_s \cdot V_k \cdot \sin\theta_{e,k-1} \\ \theta_{e,k} &= \theta_{e,k-1} + T_s \cdot \Omega_k\end{aligned} \qquad <5.19>$$

Sin embargo se podría considerar lineal de forma sencilla, simplemente haciendo la suposición de que el error de orientación θ_e .no supera en ningún caso los ±60º, ya que en ese caso se puede asumir que $\sin(\theta_e) \approx \theta_e$. Además, la entrada V se sigue considerando constante, con lo que aunque aparezca en el modelo planteado no se considera a la hora de realizar la aproximación planteada. Con dichas suposiciones, históricamente muy recurridas [Elena-99], este segundo modelo de la planta, linealizado en su formato continuo y discreto, queda de la siguiente forma:

$$\begin{bmatrix}\dot{d}_e(t)\\ \dot{\theta}_e(t)\end{bmatrix} = \begin{bmatrix}0 & V\\ 0 & 0\end{bmatrix}\begin{bmatrix}d_e(t)\\ \theta_e(t)\end{bmatrix} + \begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}V(t)\\ \Omega(t)\end{bmatrix} \quad \text{o} \quad \begin{aligned}\dot{d}_e(t) &= V(t)\cdot\theta_e(t)\\ \dot{\theta}_e(t) &= \Omega(t)\end{aligned} \qquad <5.20>$$

$$\begin{bmatrix}d_{e,k}\\ \theta_{e,k}\end{bmatrix} = \begin{bmatrix}1 & V_k\cdot T_s\\ 0 & 1\end{bmatrix}\begin{bmatrix}d_{e,k-1}\\ \theta_{e,k-1}\end{bmatrix} + \begin{bmatrix}0 & V_k\cdot T_s^2/2\\ 0 & 1\end{bmatrix}\begin{bmatrix}V_k\\ \Omega_k\end{bmatrix}$$

$$\begin{aligned}d_{e,k} &= d_{e,k-1} + V_k\cdot T_s\cdot\theta_{e,k} + \frac{V_k\cdot T_s^2}{2}\Omega_k \\ \theta_{e,k} &= \theta_{e,k-1} + T_s\cdot\Omega(t)\end{aligned} \qquad <5.21>$$

En el trabajo expuesto en este documento, no se ha utilizado el modelo anterior, puesto que sería entonces necesario conocer la relación entre el vector de salida y el de estados (ecuación de salida) y esta expresión no es fácil de obtener de forma absoluta, pues el resultado presenta bastantes no-linealidades. Sin embargo su uso se mantiene para diseñar el controlador incluido en la figura 5.4 anterior.

Con todo lo expuesto, el sistema global queda preparado para realizar las simulaciones que mostrarán los resultados de fusión obtenidos en este trabajo.

5.3.2. Simulaciones sobre el generador modelado

En este apartado final se van a mostrar distintos resultados obtenidos del algoritmo de fusión ante distintas configuraciones. Las pruebas se han agrupado en dos partes. Una primera en la que el EKF trabaja "desconectado" del sistema de navegación, y que permite comprobar su fiabilidad; y una segunda en la que sus estimaciones se incorporan en la realimentación generador de trayectorias, y se completa el objetivo de fusión para posicionamiento del robot en tareas de navegación.

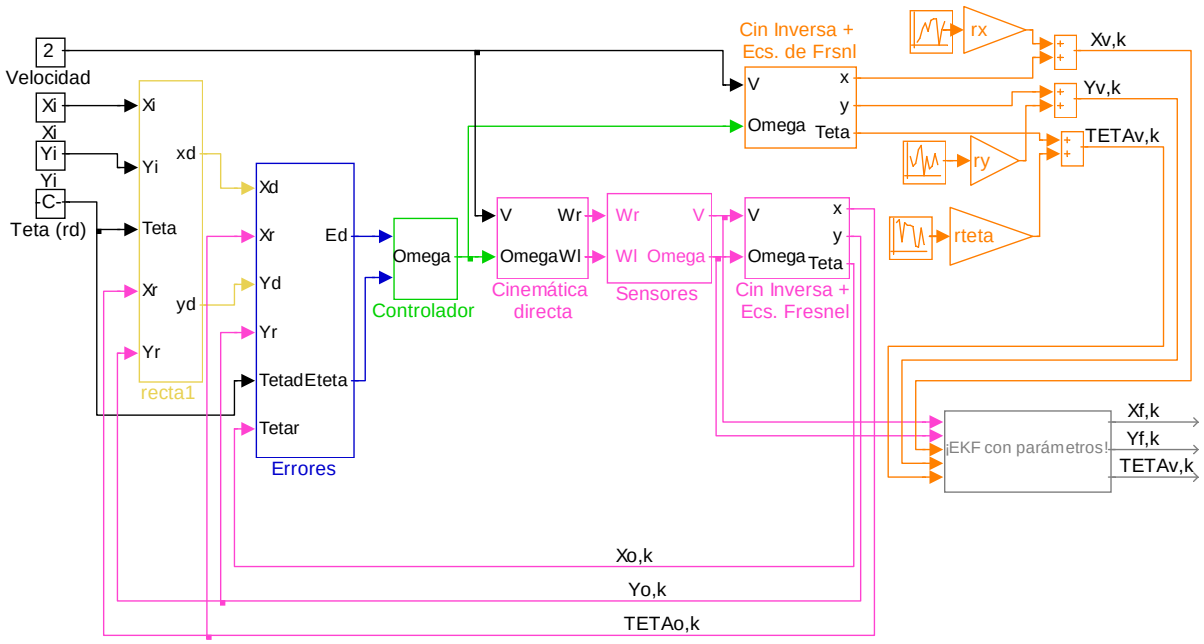


Figura 5.7. Modelo del sistema total con el EKF "desconectado" del lazo de realimentación utilizado en las primeras pruebas.

El esquema de Simulink utilizado para realizar las primeras pruebas (con el EKF "desconectado" del lazo de realimentación), es el que se muestra en la figura 5.7. Además, el color de las gráficas que se va a utilizar en todas las simulaciones es el siguiente:

- **Verde:** Consignas deseadas. Generadas por el controlador o el generador de trayectorias.
- **Azul claro:** Posición real sin ruido de ningún sistema sensorial.
- **Azul Oscuro:** Variables generadas por el sistema odométrico, con su ruido correspondientemente asociado.
- **Magenta:** Variables generadas por el sistema de visión, con su ruido correspondientemente asociado.
- **Rojo:** Salida del algoritmo de fusión EKF.

Sobre dicho diagrama se realizan las siguientes pruebas (que permiten estudiar el comportamiento del algoritmo de fusión en todas sus variantes):

1. Prueba inicial del algoritmo, con ambos sistemas sensoriales ejecutándose con el mismo periodo de muestreo y el ruido de odometría modelado sobre las velocidades angulares de las ruedas. Se asigna a R (del orden de 10e-2) y Q (del orden de 10e-4)valores coherentes con su significado. Los resultados se muestran en la figura 5.8.

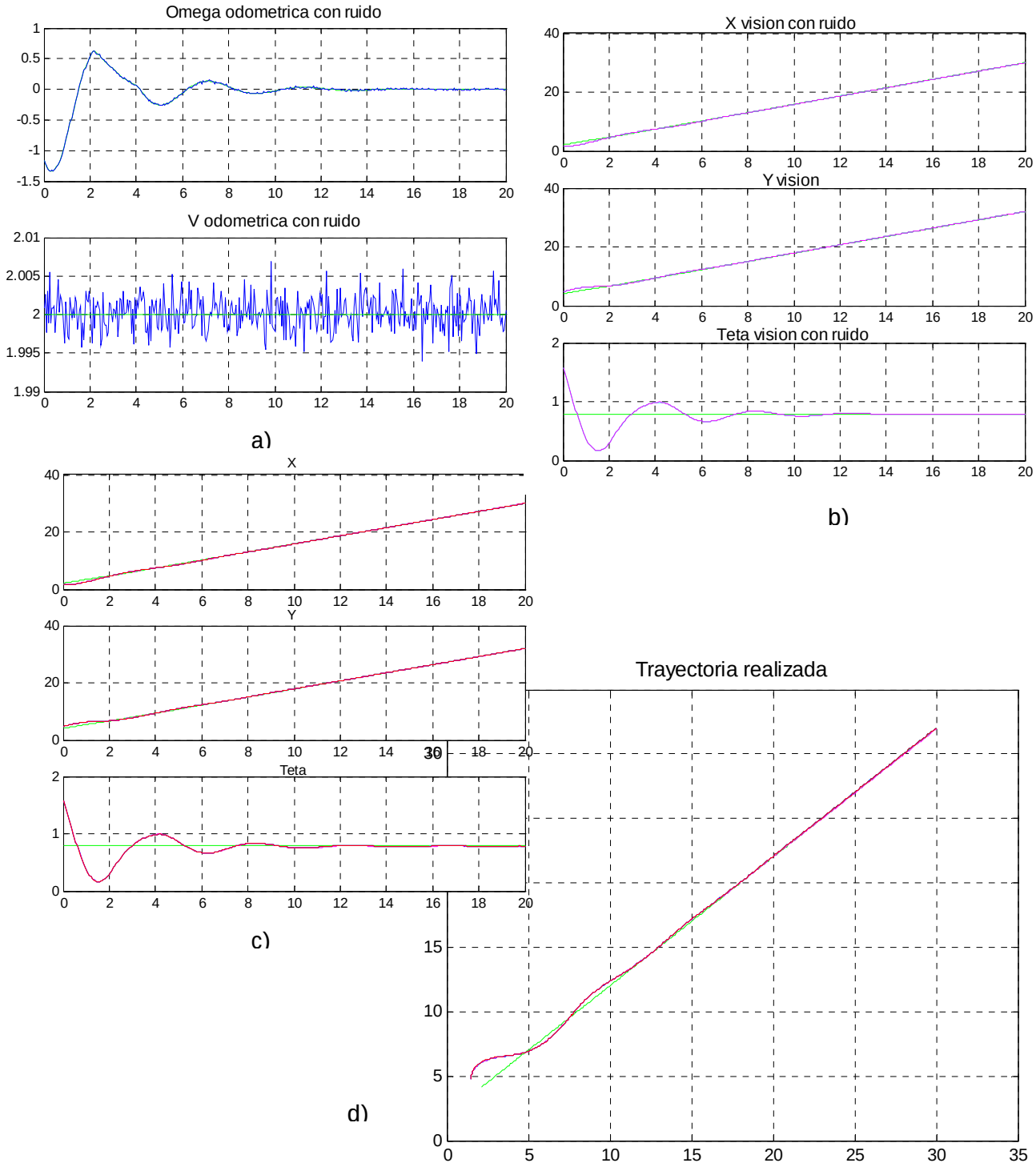


Figura 5.8. Resultados de la primera prueba (EKF "desconectado")

Tal y como se puede apreciar en la figura (gráfica d)), el estimador sigue perfectamente a la salida sin ruido (de color azul claro, que no se ve, pues aparece oculta bajo la línea roja) y el controlador puede perfectamente seguir la trayectoria sólo a partir de las medidas de odometría. Este hecho es debido a que el ruido es muy bajo, en ambos sistemas.

2. Se modifican las componentes de error, con respecto a la prueba anterior, subiendo el valor de la covarianza de cada una de ellas a valores de décimas, muy por encima de su valor real. El resultado es el que se muestra en la figura 5.9.

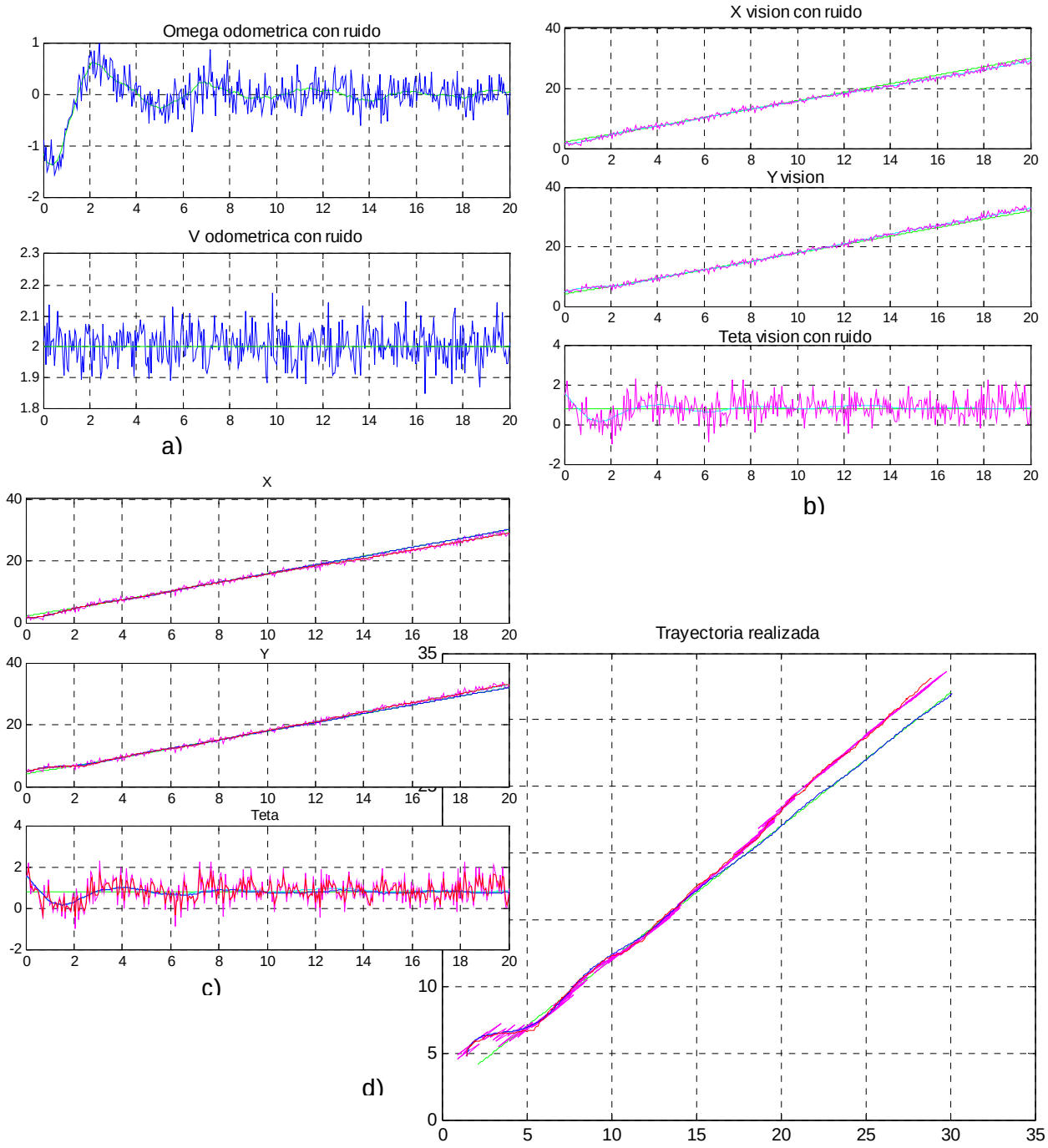


Figura 5.9. Resultados de la segunda prueba (EKF “desconectado”)

En este caso sí que se aprecia el ruido sobre las diferentes medidas (gráficas a) y b)), tanto es así, que como el controlador evoluciona con la odometría, el ruido asociado a ésta no permite que el controlador

siga sin problemas la trayectoria diseñada (en la gráfica d), las líneas azul clara y roja). Es perfectamente visible en la gráfica d) el efecto acumulativo de este ruido, que hace que la trayectoria real del robot se vaya separando cada vez más de la deseada. Por otra parte se puede apreciar también como el estimador (línea roja) se ve muy influenciado por los dos ruidos, pero en gran medida los filtra.

3. En esta prueba se desea comparar los dos modelos del ruido odométrico. Para ello, se dejan los parámetros tal y como estaban en la prueba inicial y se observa la estimación del EKF con uno y otro modelo, obteniendo los resultados de la figura 5.9.

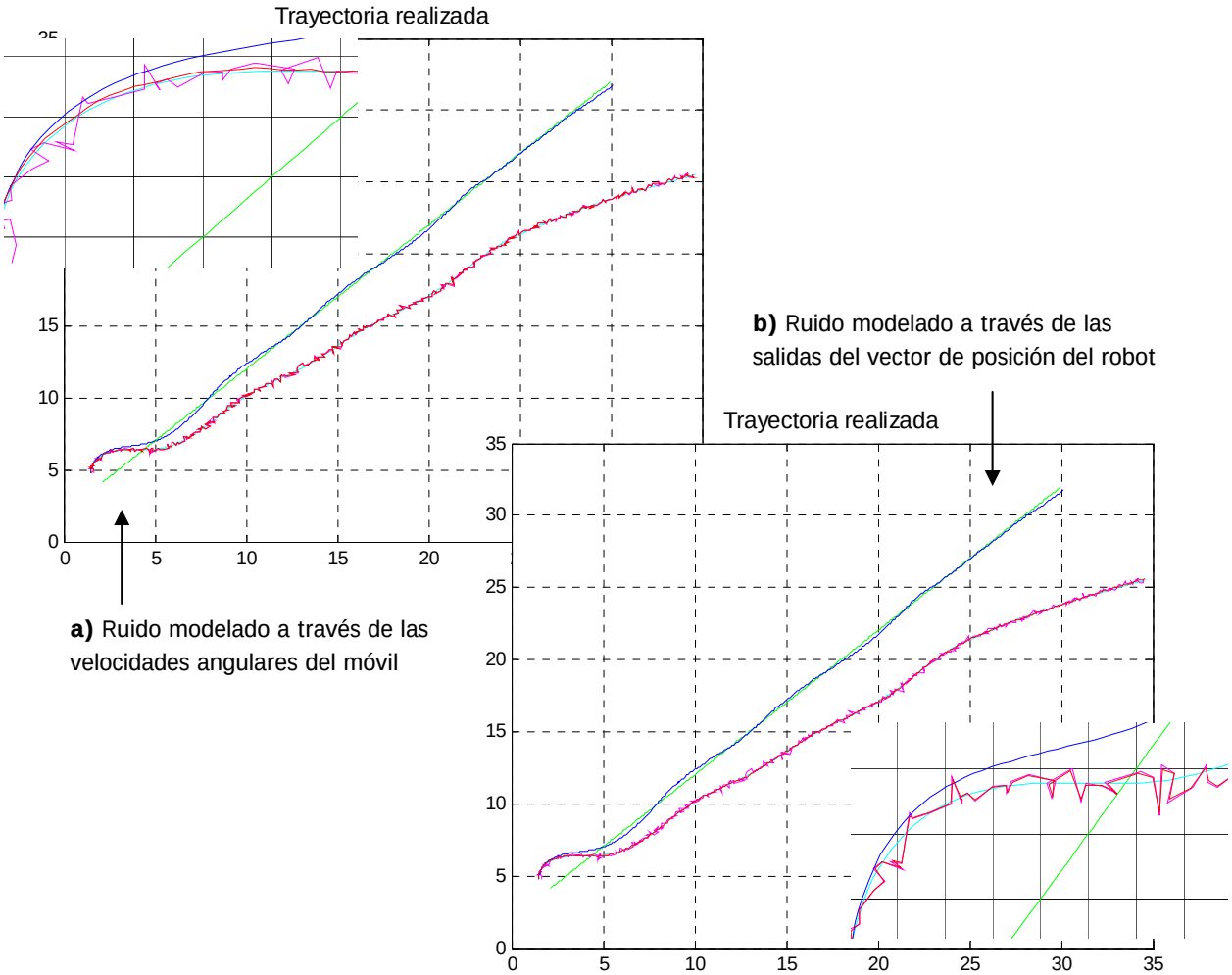


Figura 5.10. Resultados de la tercera prueba (EKF "desconectado")

Tal y como se aprecia en la figura, el modelo utilizado hasta el momento (gráfica a)) representa más fielmente al ruido odométrico. La prueba está en que el estimador (línea roja, que se ve más claramente en el segmento de gráfica ampliada) funciona mejor en el caso de la gráfica a), pues la caracterización del ruido es más acorde con la realidad de las medidas del modelo.

4. La siguiente prueba pone en práctica la diferencia de periodos de muestreo en los dos sistemas sensoriales. En este caso se configura el parámetro ‘n’ con un valor de 2, es decir, que el periodo de muestreo del sistema de visión será doble al del sistema odométrico. La configuración permanece igual a la de la primera prueba en el resto de parámetros (entre otras, el ruido odométrico vuelve a estar modelado a través de la velocidad angular de las ruedas del robot), si el ruido de visión se vuelve a introducir de alta varianza (de décimas). El resultado de la prueba se muestra en la gráfica 5.11.

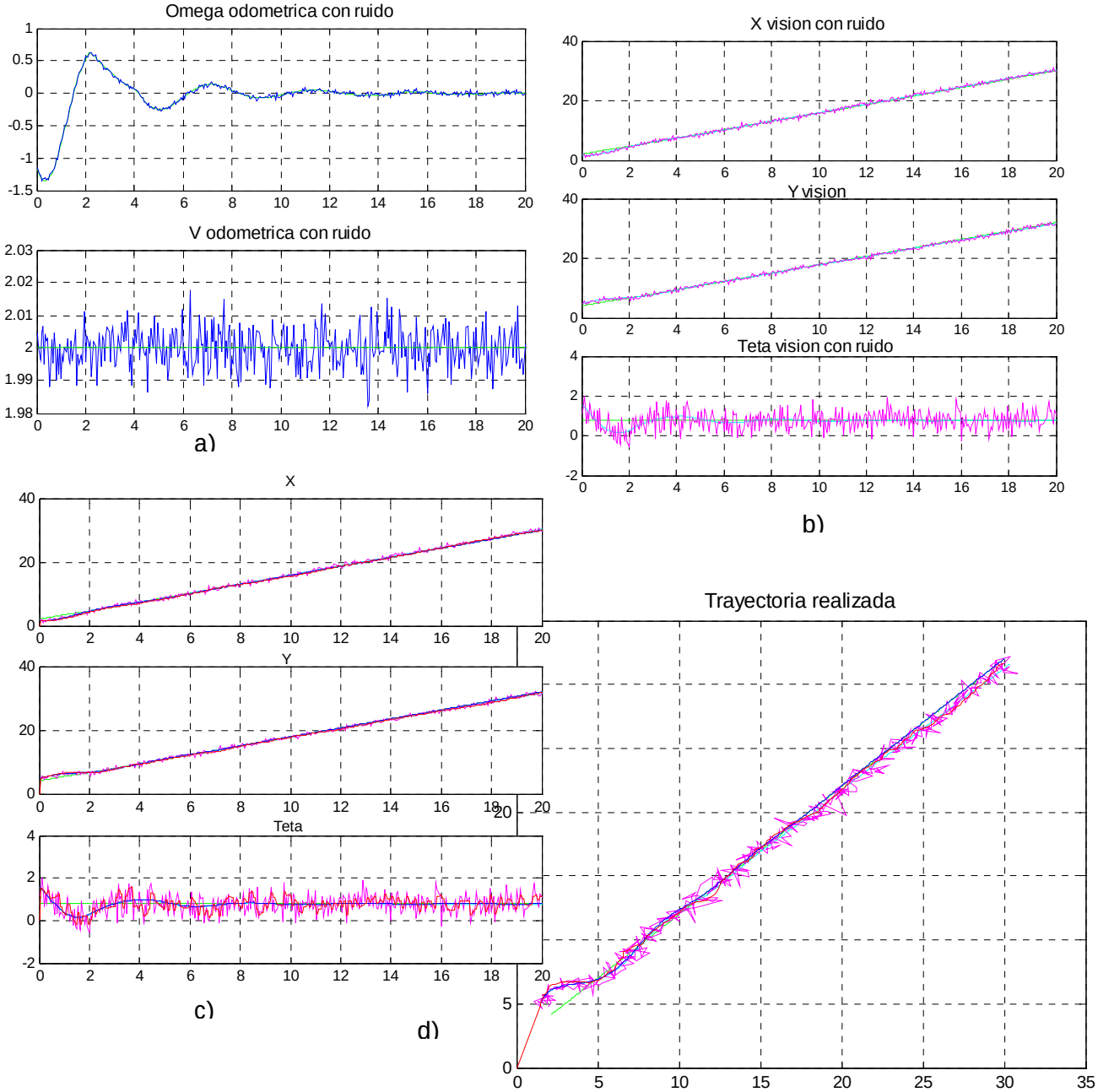


Figura 5.11. Resultados de la cuarta prueba (EKF “desconectado”)

Los resultados son evidentes a simple vista. Las medidas del algoritmo de visión son bastante malas (ver gráfica b)), pero como se integran cada dos periodos de ejecución del algoritmo de fusión no afectan tanto como en las pruebas en las se integraba todo con el mismo periodo (compárense las gráficas d) de ésta y la prueba primera, figura 5.8). Por su parte el ruido de odometría cobra más importancia ya que no se corrige en cada paso de ejecución del EKF. Este hecho se aprecia observando que con un ruido odométrico semejante al de la primera prueba (ver gráfica a) de las dos pruebas) la estimación es peor en esta última prueba.

Se han realizado diversas simulaciones modificando el parámetro ‘n’ de multiplicidad del periodo de ejecución, llegándose a la conclusión de que si este número es eleva por encima de 3 o 5 el algoritmo de fusión deja de funcionar, produciéndose una divergencia de la matriz de Kalman. Este aspecto es interesante a la hora de establecer cuál va a ser el periodo de ejecución del algoritmo SPL de visión. Teniendo en cuenta que en todos los casos se ha supuesto que T_{sv} es de 50ms (especificado por la velocidad de respuesta y el tiempo muerto de la plataforma física [López-99]) los análisis realizados sobre el parámetro de multiplicidad informan de que el periodo de ejecución del algoritmo de visión no ha de superar los 200ms, cuestión que, por otra parte, es perfectamente plausible con una plataforma apropiada para el sistema y procesado de visión, según los comentarios del autor de la tesis [Gracia-1].

Con todos los análisis mostrados se demuestra el correcto funcionamiento del sistema de fusión incluso en situaciones mucho más desfavorables que las existentes en realidad, en lo que al nivel de ruido se refiere.

Como colofón, tal y como se comentó al principio de este apartado, se ha realizado una prueba de fusión, en la que el EKF se utiliza como elemento de realimentación de la posición en lugar del sensor odométrico. Para estas pruebas se emplea el modelo que se muestra en la figura 5.12.

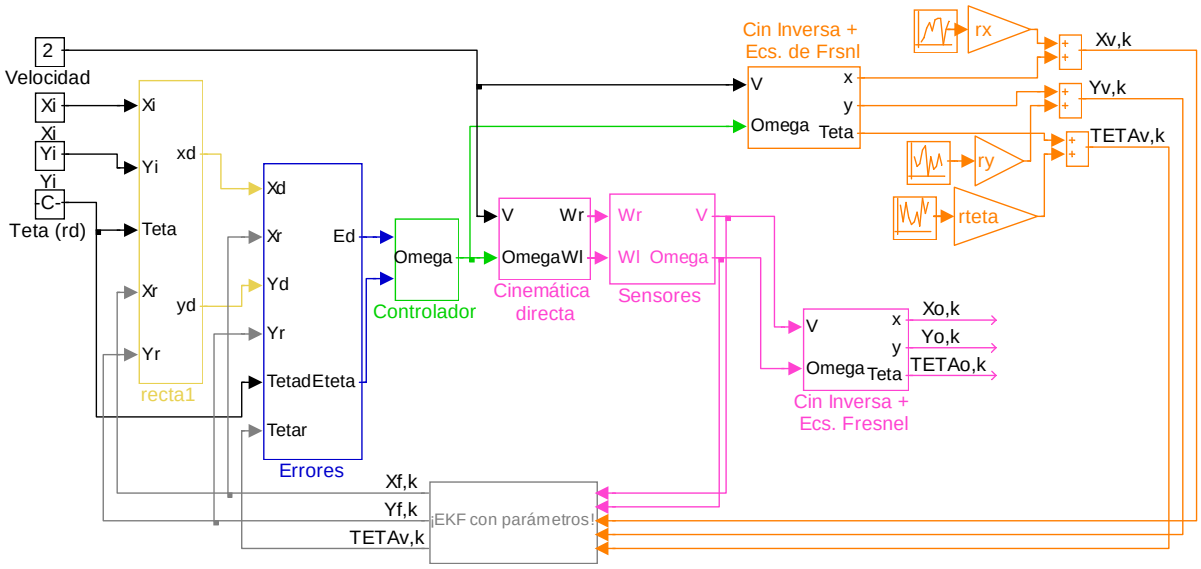


Figura 5.12. Modelo utilizado para comprobar el funcionamiento del algoritmo de fusión en el lazo de realimentación del generador de trayectorias

En este caso se realiza una única prueba, con las condiciones de ruido extremas que se probaron en el segundo ensayo, pero además añadiendo una multiplicidad de valor 2 entre los periodos de ejecución del algoritmo de visión y el de fusión. Los resultados son los que se muestran en la gráfica 5.13.

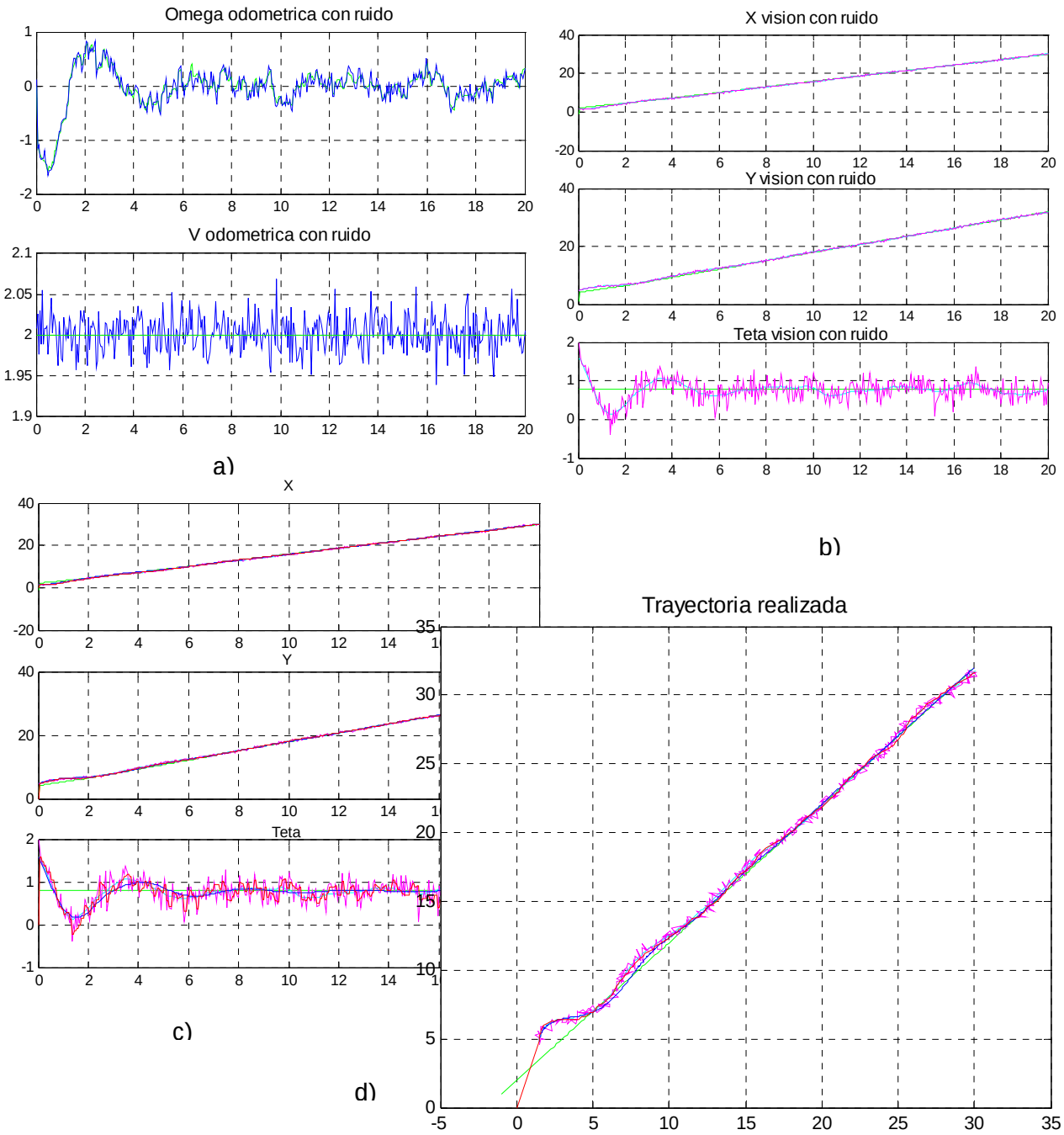


Figura 5.13. Resultados de la última prueba realizada con el algoritmo de fusión como elemento de realimentación de la posición del generador de trayectorias del robot móvil

La figura anterior muestra el éxito del trabajo de modelado e implementación de un EKF para la fusión de datos de posición procedentes de un sistema sensorial odométrico y uno de visión. Como se observa en ella, y en comparación con la figura 5.9, el algoritmo de fusión permite obtener de forma fiable la posición del robot, independientemente del nivel de ruido de las señales sensadas.

Esta conclusión valida el objetivo del trabajo planteado, y por otro lado, abre uno nuevo, que consistirá en llevar a cabo la exportación del algoritmo desarrollado a la plataforma hardware en la que se ejecuta el navegador en tiempo real.

A la hora de llevar a la práctica esto último, será necesario realizar un modelado adecuado del ruido, ya que este factor es de suma importancia en lo que a la convergencia del algoritmo se refiere, tal y como se ha demostrado en una de las pruebas realizadas. Este aspecto es sobre todo importante desde el punto de vista del sistema de visión, cuyo modelado de ruido no ha podido ser experimentado en este documento, dado que son necesarios los parámetros de salida del algoritmo SPL para implementar correctamente las matrices características.

Este y otros aspectos planteados a lo largo del documento (por ejemplo, ensayar con la técnica de Julier-Uhlmann pues el ruido supuesto gaussiano ha dejado de serlo al hacer las linealizaciones correspondientes) se plantean como posibles trabajos futuros, si bien la experiencia extraída del desarrollo de investigación me dice que en cualquier caso y teniendo cierta fiabilidad en el modelado del ruido, el EKF diseñado tiene un comportamiento excelente para la tarea concreta aquí planteada.

6. Referencias

[BarShalom&Li-93]. “Estimation and Tracking: Principles, Techniques and Software”. Ed. Artec House. 1993.

[Bonnifait&Garcia-98]. “Design and Experimental Validation of an Odometric and Goniometric Localization System for Outdoor Robot Vehicles”. IEEE Transactions of Robotics and Automation. Vol 14, Nº 4. Agosto 1998.

[Borenstein&Feng-95]. “UMBmark: A Benchmark Test for Measuring Odometry Errors in Mobile Robots”. SPIE Conference on Mobile Robots. Octubre 1995.

[Borenstein&Feng-96]. “Measurement and Correction of Systematic Odometry Errors in Mobile Robots”. IEEE Transactions of Robotics and Automation. Vol 12, Nº 5. Octubre 1998.

[Fabrizi&al.-00]. “Mobile Robot Localization via Sensor Fusion of Ultrasonic and Inertial Sensor Data”. 8th International Symposium on Robotics with Applications. 2000.

[García-01]. “Sistema de Posicionamiento y Autolocalización para Sillas de Ruedas Autónomas”. Tesis Doctoral del Dpto. Electrónica de la Universidad de Alcalá. Diciembre 2001.

[Grewal&Andrews-01] “Kalman Filtering: Theory and Practice Using MATLAB”. 2nd Ed de Cloth. ISBN: 0-471-39254-5. Enero-2001.

[Hu&Gu-99]. “Landmark-based Navigation of Mobile Robots in Manufacturing”. IEEE International Conference on Emerging Technologies & Factory Automation. Barcelona. Octubre 1999.

[Julier&Uhlmann-96]. “A New Approach for Filtering Nonlinear Systems”. Proceedings of the 1995 American Control Conference. 1996.

[Kleeman&Chong-96]. “Accurate Odometry and Error Modelling for a Mobile Robot”. MECSE. 1996.

[López-99]. “Aplicación de Controladores Óptimo y Borroso al Seguimienot d eTrayectorias de una Silla de Ruedas”. Proyecto Fin de Carrera del Dpto. de Electrónica de la Universidad de Alcalá. Marzo 1999.

[Marksacov-95]. “Mobile Vehicle Navigation in Unknown Environments: A Multiple Hypothesis Approach”. IEE Proceedings in Control Theory Applications. Vol 142. Nº 4. Julio 1995.

[Marrón-00]. “Navegación Autónoma de una Silla de Ruedas en Interiores Parcialmente Estructurados”. Proyecto Fin de Carrera del Dpto. de Electrónica de la Universidad de Alcalá. Septiembre 2000.

[Maybeck-79]. “Stochastic Models, Estimation and Control”. Vol 1. Ed. Academic Press. ISBN: 0-12-480701-1. 1979.

[McKendall-91]. “Static Decision Theory for Sensor Fusion”. 1991

[Moreno-02]. “Módulos Auxiliares para una Silla de Ruedas Autónoma”. Proyecto Fin de Carrera del Dpto. de Electrónica de la Universidad de Alcalá. Abril 2002.

[Pirjanian-98]. “Application of Voting to Fusion of Purposive Modules: An Experimental Navigation”. Elservier. Febrero 1998.

[PozoRuz-01]. “Autonomous Robot in Agriculture Tasks”. 2001.

[Sebastián-99]. “Guiado Semiautomático de una Silla de Ruedas”. Proyecto Fin de Carrera del Dpto. de Electrónica de la Universidad de Alcalá. Octubre 1999.

[Yoder&al.-96]. “Initial Results in the Development of a Guidance System for a Powered Wheelchair”. IEEE Transactions on Rehabilitation Engineering. Vol 4. Nº 3. Septiembre 1996.

[Whelch&Bishop-01]. “An Introduction to the Kalman Filter”. Universidad de Carolina del Norte. Chapel Hill. SIGGRAPH 2001.