

Pronóstico del área de contacto de los neumáticos de un vehículo vía redes neuronales recurrentes.

Leopoldo Urbina, Marco A. Moreno-Armendáriz, Carlos A. Duchanoy,
Hiram Calvo

Instituto Politécnico Nacional, Centro de Investigación en Computación,
México

Resumen. En este trabajo, un sensor virtual predictivo basado en una red neuronal recurrente es propuesto para sensar el área comprendida entre los neumáticos y el terreno, la cuál es uno de los temas más importantes en la industria automotriz, de hecho, mantener una adecuada área de contacto del neumático garantiza la comodidad y la maniobrabilidad del vehículo. El sensor suave usa una red neuronal recurrente, el entrenamiento de una red neuronal recurrente es complicado para algoritmos basados en el cálculo del gradiente, debido a la aparición de algunos valles en la superficie de error. Como alternativa, el proceso de entrenamiento se puede modelar como un problema de optimización y este es resuelto mediante un algoritmo de evolución diferencial. Este sensor virtual ha sido validado con un banco de pruebas basado en el fenómeno de reflexión interna total frustrada.

Palabras clave: Redes neuronales recurrentes, sensor virtual predictivo, evolución diferencial.

Prediction of the Tire Contact Area of a Vehicle via Recurrent Neural Networks

Abstract. In this paper, a predictive soft sensor via a Recurrent Neural Network is proposed for sensing the area between the tires and the terrain, which is one of the most important issues in the automotive industry, in fact, keeping a tire contact patch suitable ensures the comfort and maneuverability of the vehicle. The core of the soft sensor is an recurrent neural network, the training of a recurrent neural network is complicated for algorithms based on the gradient, because of the apparition of some valleys on the error surface. As an alternative, the training process is interpreted like an optimization problem and it is solved by an differential evolution algorithm. This soft sensor has been validated using a test rig based on the frustrated total internal reflection phenomenon.

Keywords: Recurrent neural networks, predictive soft sensor, differential evolution algorithm.

1. Introducción

Los neumáticos son un componente importante para garantizar la seguridad de un vehículo terrestre. La razón radica en que el área de contacto entre la llanta y el suelo es donde las fuerzas del vehículo interactúan con el entorno, mantener un área de contacto adecuada garantiza el confort y la maniobrabilidad del vehículo. Un ejemplo de la importancia de monitorear el comportamiento de la llanta es la normativa impuesta en el 2005 por la Administración Nacional de Seguridad del Tráfico (NHTSA, por sus siglas en inglés), en la cual se exige la instalación de sistemas de monitoreo de presión del neumático (TPMS, por sus siglas en inglés) [1]. Además, en el estudio presentado en [6] se demuestra que las condiciones adversas del terreno aunado con los defectos en los neumáticos, son la mayor causa de accidentes en carretera. Los sensores en los neumáticos podrían ser usados como sistemas de prevención, también para mejorar la maniobrabilidad en condiciones de terreno desfavorables, además de ser parte de un sistema de control, como pueden ser las suspensiones adaptativas, sistemas de frenado anti bloqueo, sistemas avanzados de asistencia al conductor o prevención de colisiones. El desarrollo de los sensores del neumático presenta varios retos, los principales están reportados en [24]. El primer reto es desarrollar un proceso de medición directa para aplicaciones reales, el segundo es crear un sistema de comunicación inalámbrica entre la llanta y el vehículo que trabaje sin batería y el último es caracterizar y probar el sensor en condiciones reales.

En [21] se presentan algunas metodologías usadas para medición de distintos parámetros en el auto, como son presión y temperatura en los neumáticos, entre otros; del mismo modo se reportan metodologías para detección de fallas de un vehículo terrestre. Un primer intento al medir el área de contacto fue mediante la obtención de una imagen térmica de la llanta, en esta se podrán observar el área que está en contacto con el suelo, debido al calor generado por la fricción con el terreno, esta técnica presenta algunas desventajas, por ejemplo, es necesario que pase cierto tiempo para que la temperatura de la llanta aumente. Algunas alternativas para medir directamente el área de contacto son mediante un sensor piezoeléctrico, estas están reportadas en [20], [23], [25], [21] y [22]. Los trabajos previamente mencionados tienen algunos problema en común, debido a que los sensores se encuentran dentro del neumático, si el sensor se daña, se necesitan herramientas especializadas para el remplazo, además la manufactura requerida para el desarrollo del sensor es costosa. Una alternativa es obtener la medición de manera indirecta, por ejemplo, en [27] se usa un sistema de posicionamiento global (GPS, por sus siglas en inglés) para estimar los ángulos de deslizamiento longitudinal, del mismo modo en [4], se logra estimar la rigidez en los extremos del dibujo de una llanta mediante la información de velocidad en el GPS en conjunto con el sistema de navegación inercial (INS, por sus siglas en inglés).

Todos los sistemas físicos reales contienen parámetros que son difíciles de medir, ya sea que el costo del sensor sea elevado, el proceso de medición requiera de un largo tiempo o la medición requiere de un ambiente controlado. Con el fin de estimar las mediciones de dichas variables difíciles de medir, se desarrollaron los sensores virtuales. Un sensor virtual es un modelo computacional el cual permite la medición de casi cualquier parámetro en tiempo real [11]. El sensor virtual relaciona mediciones más sencillas de obtener con la medición de interés, todo a través de un modelo. Los sensores virtuales se dividen en, manejados por datos y manejados por modelo. Los sensores manejados por modelo, requieren forzosamente de un modelo matemático que describa el sistema. En la mayoría de los casos la dinámica del sistema es muy compleja para poderla modelar [29]. Los sensores virtuales manejados por datos no necesitan de un modelo matemático, estos sensores forman relaciones entre otras mediciones más fáciles de obtener y la medición deseada. Los sensores virtuales basados en datos requieren de un modelo de caja negra para simular el sistema, tales como, métodos estadísticos multivariantes, maquinas de soporte vectorial, redes neuronales. Por ejemplo, en [33], se propone un sensor virtual basado en el filtro de Kalman y modos deslizantes, para estimar el centro de gravedad y el ángulo de deslizamiento lateral del vehículo. En [26] se diseña un sensor virtual basado en una red neuronal con inversión izquierda, capaz de estimar los estados del vehículo tales como la velocidad de guiñada y el ángulo de deslizamiento lateral, estados que son importantes para el control de estabilidad del vehículo. En [7] se desarrolla un sensor virtual mediante la aplicación de un filtro de Kalman extendido, para estimar el coeficiente de fricción y el ángulo de deslizamiento lateral.

Las redes neuronales son usados comúnmente en los sensores virtuales basados en datos debido a sus capacidades como aproximadores universales [17]. Dependiendo la arquitectura y el tipo de conexiones que tenga una red neuronal, estas pueden ser aplicada para resolver problemas con dinámica compleja, una red neuronal recurrente (RNN, por sus siglas en inglés) puede resolver problemas modelados como series de tiempo. Por lo tanto un sensor virtual que use una red neuronal recurrente, además de poder estimar mediciones complejas de un sistema dinámico, es capaz de predecirlas. Un ejemplo de sensor predictivo se presenta en [12], donde se usa un modelo de inferencia neuro-difuso adaptativo para predecir el comportamiento de un vehículo en la maniobra de cambio de carril. En [31] se presenta un sistema de control inteligente de neumático, sensores piezoeléctricos se usan como entrada a una red neuronal la cual estima y predice, las diferentes fuerzas presentadas en el neumático.

Una de las principales problemáticas en el uso de redes neuronales es el entrenamiento. El proceso de entrenamiento consiste en ajustar los pesos sinápticos y bias de la red hasta que todos los conjuntos de entrada de la red estén relacionados con los valores deseados a la salida. Existen distintos métodos con los cuales la red puede ser entrenada, la mayoría de estos métodos tratan de calcular el gradiente o en el mejor caso, aproximarlo, el algoritmo de Levenberg-Marquardt [30] es uno de ellos. Estos métodos son eficientes para el entrenamiento de redes neuronales sin retroalimentación, en el caso de las redes recurrentes existen

diversas dificultades [2]. La principal razón es la presencia de valles espurios en la superficie del error; estos valles son mínimos locales y cambian dependiendo de la secuencia de entrada [8]. En [28] se describe una forma de lograr un entrenamiento exitoso sin caer en un valle espurio, basta con monitorear la magnitud del gradiente, si este es muy grande, es un indicio de que estamos en un valle, entonces el conjunto de entrada se cambia temporalmente para entrenar con otra secuencia. Existen otros métodos para lograr un entrenamiento exitoso, en [3] sugieren usar algoritmos de búsqueda estocástica, tales como el algoritmo de recocido simulado, evolución diferencial, entre otros. El problema de entrenamiento puede ser interpretado como un problema de optimización, por lo tanto puede ser resuelto por métodos no basados en el gradiente [15]. En [19] es usado un algoritmo de evolución diferencial como base de entrenamiento de una red neuronal, estos métodos estocásticos pueden evitar perderse en los valles espurios debido a su mecanismo de búsqueda.

En este trabajo se propone una aplicación de una red neuronal recurrente en un sensor virtual predictivo. Este sensor sería útil en un sistema de suspensión activa ya que el sistema podría predecir perturbaciones. Además se propone un arreglo de sensores que nos permita medir un conjunto de variables que están relacionados con el área de contacto del neumático.

2. Presentación del problema

El área de contacto entre el neumático y el terreno es el último y el más importante eslabón en la cadena del sistema de propulsión, debido a que es donde todas las fuerzas externas interactúan con el vehículo. Uno de los pasos más importantes en el desarrollo de un sensor virtual es elegir el conjunto de variables físicas que estén estrechamente relacionadas con el área de contacto de la llanta, la razón radica en que una vez elegidas, se elaborará toda la instrumentación electrónica para sensar estas variables. A pesar del desarrollo en el campo de los sensores virtuales, no se ha encontrado trabajo alguno en el cual se use un sensor virtual para estimar el área de contacto de la llanta. Existen distintas metodologías para obtener de manera directa esta área, en [14] se estima el área de contacto por medio de figuras geométricas, por otra parte, en [32] se expone un modelo de elemento finito para calcular la misma área. El vehículo en el que el sensor virtual será probado es el Baja 5sc SS de hpi-racing® [18]. El Baja 5sc SS es un vehículo de radio control a escala 1/5, fabricado con un motor de 2.9 HP de dos tiempos, este vehículo posee todos los sistemas de un auto real. Se escogió este vehículo debido a la facilidad con la que se puede instrumentar e implementar los sensores, además gracias a las dimensiones del vehículo, este puede ser sometido a pruebas en el banco de pruebas desarrollado, por último, este vehículo no requiere de equipo especial para realizar ciertos ajustes.

3. Diseño del sensor predictivo

En [11], Fortuna propone 4 pasos para el diseño general de un sensor virtual, en este trabajo se ajustaron estos pasos para nuestra aplicación.

3.1. Selección de variables

El primer paso en el diseño de un sensor virtual es el análisis de los datos disponibles, el resultado de este análisis es una lista de variables que estén estrechamente relacionadas con el área de contacto del neumático. A partir de esta selección, se diseñará la instrumentación necesaria. Como apoyo, se usó un modelo matemático previamente desarrollado por nuestro grupo en [9]. El modelo matemático fue desarrollado para un vehículo distinto al propuesto, de cualquier modo el modelo nos ayudó a reconocer las variables clave que afectan en mayor medida al área de contacto. El resultado del estudio fue el siguiente conjunto de variables, la aceleración de las llantas en el eje z (a_{t_1} , a_{t_2} , a_{t_3} , a_{t_4}), el desplazamiento de la suspensión en cada amortiguador (x_{s_1} , x_{s_2} , x_{s_3} , x_{s_4}), el ángulo de cabeceo y balanceo lateral del chasis (Φ y Θ respectivamente) y por último el ángulo de la dirección (Ω). Para futuras referencias el número de cada llanta empieza de la frontal izquierda y continua en contra del sentido de las manecillas del reloj.

3.2. Instrumentación y adquisición de datos

La instrumentación electrónica debe poder ser capaz de sensar las variables físicas que se requieren, a pesar de las vibraciones del terreno o del motor. El primer conjunto de mediciones son los desplazamientos de la suspensión en cada amortiguador. Los sensores escogidos fueron los sensores de flexión de SpectraSymbol®. La razón por la que se eligió este sensor es que no interfiere en el movimiento natural de la suspensión ya que la fuerza necesaria para doblar el sensor es despreciable comparado con la fuerza en el amortiguador. El sensor se coloca entre el chasis y la llanta, a medida que el sensor se deforma, este cambiará su resistencia interna, por lo tanto estos sensores requieren de un circuito que transforme los cambios de desplazamiento en diferencia de potencial. El segundo conjunto de mediciones son las aceleraciones. Se seleccionó el acelerómetro ADXL345 de Sparkfun® debido a que cumplía los requerimientos de la instrumentación. Para leer el acelerómetro solo es necesario orientar el sensor a los ejes del sistema y establecer un protocolo de comunicación, normalmente este es establecido por medio de un microcontrolador. Los ángulos de cabeceo y balanceo lateral del chasis pueden ser leídos por un giroscopio, pero se eligió una unidad de masa inercial (IMU, por sus siglas en inglés), una IMU puede sensar aceleración, orientación y posición angular por cada eje. La IMU que cumple los requerimientos de la instrumentación es la Razor IMU de Sparkfun®, al igual que el acelerómetro, antes de empezar a sensar, es necesario colocar la IMU orientada con los ejes del chasis. La última variable es el ángulo de la dirección. El proceso del vehículo para realizar un cambio en la dirección es el siguiente,

el piloto indica el cambio usando un transmisor de radio frecuencia, el vehículo recibe la señal mediante un receptor de radio frecuencia y es convertida en una señal de modulación por ancho de pulsos (PWM, por sus siglas en inglés), el PWM modifica la posición de un servomotor y este a su vez por medio de la barra de dirección modifica el ángulo de la dirección. El punto en el que se puede sensar el ángulo sin alterar o modificar este proceso es copiando la señal de PWM por medio de un contador o timer.

Para finalizar la instrumentación se requiere de un microcontrolador que lea y administre los datos de los sensores. Se eligió el microcontrolador Tiva TM4C123GXL de Texas Instruments® debido a que cuenta con los suficientes módulos para interpretar la señal de los sensores de flexión, del mismo modo puede codificarse el protocolo de comunicación entre los acelerómetros y la IMU además tiene un módulo para capturar la señal PWM del receptor. El principal reto de la instrumentación es asegurar que todas las mediciones sean tomadas en el mismo tiempo, o con un retraso mínimo.

3.3. Medición del área de contacto del neumático

El banco de pruebas para medir el área de contacto de las llantas está basado en el fenómeno de reflexión interna total, el cual ocurre cuando la luz incide sobre un medio con menor índice de refracción (del vidrio al aire), el rayo se desvía de la normal y para cualquier ángulo de incidencia menor que el ángulo crítico, parte de la luz incidente será transmitida y parte será reflejada. Ver [10] para más información sobre el fenómeno. La tecnología basada en la reflexión interna total frustrada (FTIR, por sus siglas en inglés), añade una superficie extra para interactuar con el medio óptico. El banco de pruebas basado en la tecnología FTIR consta de un panel de acrílico con leds infrarrojos alrededor de su periferia, los leds administran la luz infrarroja dentro del panel. Cuando una neumático toca el panel de acrílico, debido al mayor índice de refracción causado por el neumático, la luz infrarroja es reflejada del área que está en contacto con el panel. Debajo del panel, el banco de pruebas tiene instalada una cámara para capturar los rayos infrarrojos reflejados. La imagen tomada por la cámara pasa por un filtro pasa bandas, con el fin de solo quedarnos con el espectro infrarrojo, que es el área en contacto con el panel; después la imagen pasa por un proceso de agrupación y conteo, el cual consiste en separar las áreas de cada llanta y contar los pixeles de cada área. De este modo se obtiene la medida del área de contacto de la llanta.

3.4. Red neuronal recurrente como parte del sensor virtual

Existen distintos tipos de métodos para aproximar un modelo de caja negra, algunos de ellos son; maquinas de soporte vectorial, métodos estadísticos multivariantes, métodos adaptativos, redes neuronales y métodos híbridos. En este trabajo se eligió el modelo de redes neuronales recurrentes debido al gran desempeño que han tenido modelando sistemas dinámicos complejos, además del desempeño en la predicción de diversos sistemas [5]. La capacidad de predicción

de una red neuronal recurrente es debido a sus conexiones de retroalimentación, estas conexiones permiten a la RNN aprender series de secuencias. La arquitectura más usada de una RNN consta de un perceptrón multicapa con ciclos de retroalimentación. Para el diseño del sensor virtual se propone una red neuronal no lineal autorregresiva con entrada exógena (NARX, por sus siglas en inglés). La arquitectura de la NARX es la siguiente, necesitamos 11 entradas (variables elegidas) más un bloque de retraso de 3 tiempos, 3 entradas de retroalimentación de la salida (las 4 áreas de contacto), esto nos da un total de 56 entradas que van están conectadas a una capa oculta de 20 neuronas; la capa de salida consta de las 20 entradas de la capa anterior, más 3 entradas de retroalimentación de la salida (las 4 áreas de contacto), dando un total de 32 entradas que se conectan finalmente a 4 neuronas.

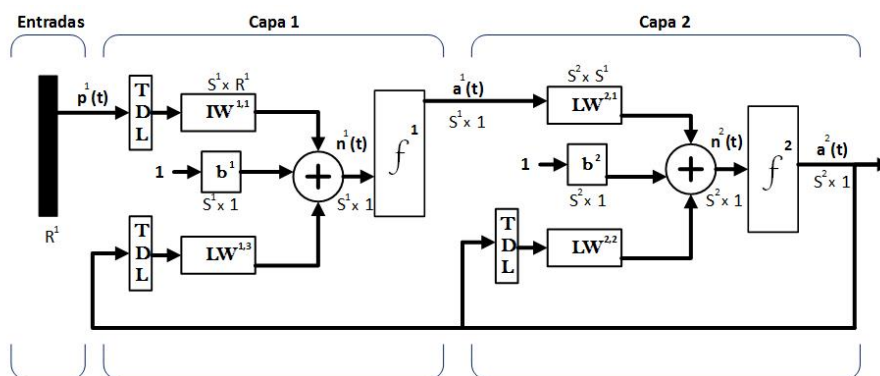


Fig. 1. Red neuronal NARX

Esta tipo de red puede ser entrenada usando algoritmos basados en el cálculo del gradiente (Ver [13]). Estos métodos presentan dificultades debido a la presencia de valles espurios en la superficie del error (Ver [16]). Estos valles son ocasionados por el conjunto de entrada y no tienen relación alguna con el mínimo real. Este problema puede ser considerado como un proceso de optimización, dónde los valles espurios pueden ser interpretados como mínimos locales. Dado este planteamiento, los métodos de optimización estocástica pueden presentar una gran ventaja respecto a los métodos basados en el gradiente, esto es debido al mecanismo de exploración que los métodos estocásticos presentan, esto les permite sortear los mínimos locales y no perderse en ellos. En este trabajo, seleccionamos como método de entrenamiento una versión modificada del algoritmo de entrenamiento basado en evolución diferencial que fue propuesto inicialmente para redes con propagación hacia adelante en [19], el cual se aplicó a una red neuronal recurrente.

Entrenamiento con el algoritmo de evolución diferencial El algoritmo de evolución diferencial puede ser clasificado como un algoritmo decodificador evolutivo ideal para espacios de búsqueda continua. Por esta razón, este puede ser aplicado para optimizar los pesos y bías de una red neuronal recurrente. La salida de la red neuronal recurrente está en función a los pesos y bías de cada capa k . Ver ecuación 1.

$$L(k) = f(w_{K,i,j,Q} * Int_K + b_{K,i,Q}), \quad (1)$$

donde $w_{K,i,j,Q}$ es la matriz de pesos, Int_K es el vector de entrada, $b_{K,i,Q}$ es el vector de bías, $f(*)$ es la función de transferencia y $L(k)$ es la salida de la capa.

En el entrenamiento clásico, las entradas de la primera capa de la red y la salida son datos conocidos, los pesos y bías de la red son ajustados con el fin de obtener un mapeo del conjunto de entrada con el de salida. El ajuste de pesos y bías se lleva a cabo reduciendo al mínimo la función de error de la red (ecuación 2).

$$E = \frac{1}{2}(L(F) - T_g)^2, \quad (2)$$

donde T_g es el salida deseada y $L(F)$ es la salida actual de la red.

De este modo, el problema de entrenamiento es ahora un problema de optimización en el que el objetivo es minimizar la función de error al ajustar los valores de pesos y bías de la red. Como la mayoría de algoritmos evolutivos, el algoritmo de evolución diferencial propone una población de candidatos a solución, no solo una única solución. El algoritmo de evolución diferencial usa un esquema de reproducción de la población diferente de otros algoritmos evolutivos. Después de la inicialización de la primera población, los vectores de población de las nuevas generaciones son tomados al azar y combinados para crear nuevos vectores mutados de pesos y bías (ecuación 3 y 4), a este paso se le conoce como mutación.

$$u_{K,i,j} = w_{K,i,j,R1} - F * (w_{K,i,j,R2} - w_{K,i,j,R3}), \quad (3)$$

$$v_{K,i,j} = b_{K,i,R1} - F * (b_{K,i,R2} - b_{K,i,R3}), \quad (4)$$

donde F es la constante de mutación.

Los vectores mutados son combinados con el vector original usando un factor de cruce CR , el cual generará un nuevo candidato, a esta etapa se le conoce como mutación. Finalmente, el desempeño del individuo original y el nuevo candidato a solución son comparados y se elimina al que presente el peor desempeño, quedándose el mejor individuo para la siguiente generación (Ver algoritmo 1).

4. Experimentos y resultados

4.1. Condiciones del experimento

El banco de pruebas construido permite una medición precisa del área de contacto del neumático. El principal inconveniente es que no permite realizar

Algorithm 1 Algoritmo de entrenamiento basado en evolución diferencial

```

1: Mientras La condición de termino no sea alcanzada Hacer
2:   Para Cada elemento  $Q$  de la población Hacer
3:     Seleccionar Números enteros aleatorios  $R1, R2, R3 \in [1, Población]$ 
4:     Para Cada capa  $K$  de la red neuronal Hacer
5:       Mutar cada  $i, j$  peso de la red
6:        $u_{K,i,j} = w_{K,i,j,R1} - F * (w_{K,i,j,R2} - w_{K,i,j,R3})$ 
7:       Mutar cada  $i$  bias
8:        $v_{K,i,j} = b_{K,i,R1} - F * (b_{K,i,R2} - b_{K,i,R3})$ 
9:       Cruzar cada  $i, j$  peso de la red
10:      Si  $Rand(0, 1) \leq CR$ 
11:         $w_{newK,i,j,Q} = u_{K,i,j}$ 
12:      De otro modo
13:         $w_{newK,i,j,Q} = w_{K,i,j,Q}$ 
14:      Fin Si
15:      Cruzar cada  $i$  Bias
16:      Si  $Rand(0, 1) \leq CR$ 
17:         $b_{newK,i,Q} = v_{K,i,Q}$ 
18:      De otro modo
19:         $b_{newK,i,Q} = b_{K,i,Q}$ 
20:      Fin Si
21:    Fin Para
22:  Para Cada secuencia de entrenamiento Hacer
23:    Para Cada capa  $K$  de la red neuronal
24:      Evaluar la RNN usando los valores anteriores de pesos y bias
25:       $Lo_k = f(w_{K,i,j,Q} * Int_K + b_{K,i,Q})$ 
26:       $fitO_k = \frac{1}{2}(Lo - T_g)^2$ 
27:      Evaluar la RNN usando los valores nuevos de pesos y bias
28:       $Ln_k = f(w_{newK,i,j,Q} * Int_K + b_{newK,i,Q})$ 
29:       $fitN_k = \frac{1}{2}(Ln - T_g)^2$ 
30:    Fin Para
31:    Seleccionar las soluciones y compararlas con el objetivo
32:    Si  $sum(fitO_k) \geq sum(fitN_k)$ 
33:       $b_{K,i,Q} = b_{newK,i,Q}$ 
34:       $w_{K,i,j,Q} = w_{newK,i,j,Q}$ 
35:    De otro modo
36:       $b_{K,i,Q} = b_{K,i,Q}$ 
37:       $w_{K,i,j,Q} = w_{K,i,j,Q}$ 
38:    Fin Si
39:  Fin Para
40: Fin Para
41: Fin Mientras

```

mediciones del auto en movimiento. Por lo tanto, se desarrollaron una sucesión de movimientos que simulan una pista de prueba, esta sucesión contiene maniobras de giro, lapsos de aceleración y frenado.

4.2. Entrenamiento de la red neuronal recurrente

El entrenamiento de nuestra RNN consiste en ajustar los parámetros de la RNN (pesos y bías) por medio del algoritmo de evolución diferencial expuesto en la sección 3.4. El entrenamiento se considera exitoso si el conjunto de variables de entrada mapea al valor de salida deseado. La precisión del sensor virtual aumentará, si el conjunto de entrenamiento contiene todos los movimientos posibles en el vehículo. Por esta razón es primordial simular y sensar todos estos movimientos en el banco de pruebas.

La arquitectura de la red neuronal recurrente NARX es la siguiente, 11 entradas (variables elegidas) más un bloque de retraso de 3 tiempos (total de 44 entradas), más 3 entradas de retroalimentación de la salida (las 4 áreas de contacto), esto nos da un total de 56 entradas que van están conectadas a una capa oculta de 20 neuronas ($S^1 = 20$); la capa de salida consta de las 20 entradas de la capa anterior, más 3 entradas de retroalimentación de la salida (las 4 áreas de contacto), dando un total de 32 entradas que se conectan finalmente a 4 neuronas ($S^2 = 4$). Para el proceso de entrenamiento se realizaron 4 experimentos cada uno con distintas condiciones de manejo y terreno, cada uno de estos experimentos contiene un total de mil datos de medición. El horizonte de predicción propuesto en este trabajo es de $33ms$, este tiempo es mayor al tiempo de respuesta de una suspensión activa. El horizonte de predicción elegido es la diferencia entre anticiparse a una perturbación o reaccionar a ella. Para el algoritmo de evolución diferencial se uso un factor de cruce $CR = 0.5$, un valor de constante de mutación $F = 0.5$. Bajo estos criterios el entrenamiento se realizó de manera exitosa. Una vez ajustados los parámetros de la RNN esta puede ser embebida en cualquier hardware capaz de recibir las señales de los sensores para así predecir el área de contacto.

4.3. Resultados y comparaciones

Una vez entrenada la RNN, con el objetivo de evaluar el desempeño del sensor virtual predictivo, se realizó una secuencia de movimientos de prueba, estos movimientos fueron sensados tanto por la instrumentación como por el banco de pruebas, el objetivo de la prueba es comparar los resultados obtenidos por la RNN y el banco de pruebas. Cada experimento de prueba consta de mil datos de medición tanto del banco de pruebas como del sensor virtual. En la figura 2 se pueden observar los comparación entre las mediciones del sensor virtual y las del banco de pruebas. Dónde la línea punteada es la medición obtenida por el sensor virtual y la línea continua es la medida obtenida por el banco de pruebas.

Se puede notar que la señal predicha es casi idéntica a la medida real, la medida predicha presenta un desfase de $33.33ms$ que es el horizonte de predicción que se había escogido previamente, la tabla 1 contiene los resultados promedio de las pruebas realizadas a cada llanta. De manera general, el error obtenido entre el sensor virtual y el banco de pruebas es aceptable dado que en promedio, el porcentaje de error es menor al 10% que es aproximadamente $75mm^2$.

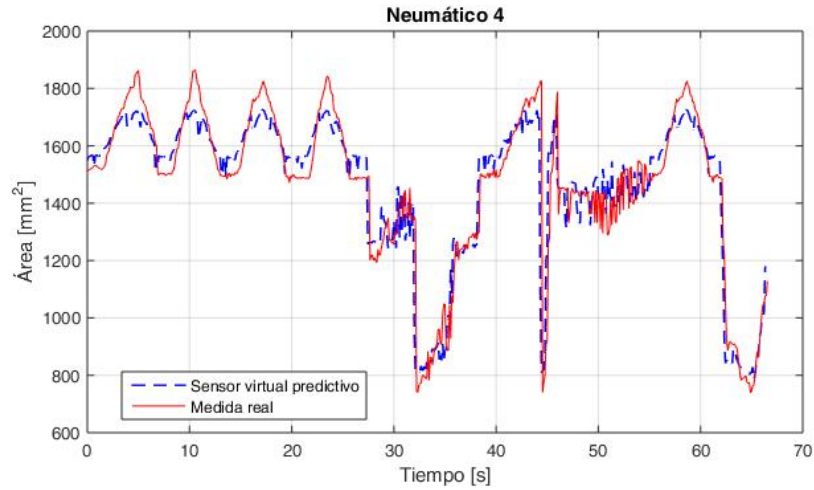


Fig. 2. Comparación entre las mediadas del banco de pruebas y el sensor virtual de la llanta 4

Tabla 1. Experimentos de prueba

Promedios	Llanta 1	Llanta 2	Llanta 3	Llanta 4
Promedio del error de predicción (mm^2)	22.32	7.25	32.28	25.36
Porcentaje promedio del error de predicción (%)	9.0997	5.8069	5.4906	4.0049

4.4. Discusión

Elegir un horizonte de predicción influye en la precisión de la predicción. Si se elige un horizonte de predicción muy amplio, la predicción es más susceptible a fallar. Por otro lado, pensando en una aplicación real, nuestro problema está acotado al tiempo de respuesta del amortiguador magnetoreológico en una suspensión activa, que para nuestro caso es menor a $33.33ms$. El horizonte de predicción propuesto permitirá el correcto funcionamiento de la suspensión activa. En la figura 2 se puede apreciar que el área predicha por el sensor virtual es muy similar al área medida una vez alcanzado el horizonte de predicción. Esto demuestra que el horizonte escogido es funcional, ya que predice de manera correcta además de permitir el correcto funcionamiento de una suspensión activa. Los resultados de la tabla 1 indican que el porcentaje promedio del error es de 9.0997%. El sensor virtual puede predecir el área de contacto de los neumáticos de manera precisa.

5. Conclusiones

Este trabajo realizó un sensor virtual predictivo mediante redes neuronales recurrentes capaz de predecir el área de contacto de los neumáticos $33.33ms$ en el futuro. Este tiempo permite una predicción precisa del área, así como la activación de una suspensión activa, esto con el fin de anticiparse a las perturbaciones, en lugar de reaccionar a ellas. El problema de entrenar una red neuronal recurrente por métodos basados en el cálculo del gradiente se debe a la presencia de valles espurios. Debido a la capacidad de búsqueda de los algoritmos estocásticos, estos son una excelente alternativa para el problema. Se realizó la instrumentación electrónica necesaria para sensar las variables físicas seleccionadas, de igual modo se diseñó y construyó un banco de pruebas capaz de medir el área de contacto de las llantas. Por último se probó la eficiencia del sensor virtual predictivo demostrando tener una porcentaje de error promedio menor al 10%

6. Trabajo a futuro

Con el deseo de continuar con el desarrollo alcanzado por este trabajo, se proponen las siguientes ideas: Seleccionar el menor número de variables de entrada pero que a la vez tengan una mayor relación con el área de contacto de la llanta, usar el sensor virtual como parte de un sistema de suspensión activa, finalmente, llevar la metodología a un vehículo convencional.

Agradecimientos. Los autores agradecen a los revisores por sus sugerencias, las cuales ayudaron a mejorar la calidad de la investigación, del mismo modo agradecemos el apoyo del Instituto Politécnico Nacional (SIP-IPN, COFAA-IPN, BEIFI-IPN) y del gobierno mexicano (SNI y CONACYT).

Referencias

1. Administration, N.H.T.S., et al.: Federal motor vehicle safety standards; tire pressure monitoring systems; controls and displays. Tech. rep., Technical report, Department of Transportation, <http://www.nhtsa.gov/cars/rules/rulings/tirepresfinal/index.html> (2000)
2. Atiya, A.F., Parlos, A.G.: New results on recurrent network training: unifying the algorithms and accelerating convergence. *Neural Networks, IEEE Transactions on* 11(3), 697–709 (2000)
3. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on* 5(2), 157–166 (1994)
4. Bevly, D.M., Sheridan, R., Gerdes, J.C.: Integrating ins sensors with gps velocity measurements for continuous estimation of vehicle sideslip and tire cornering stiffness. In: *American Control Conference, 2001. Proceedings of the 2001.* vol. 1, pp. 25–30. IEEE (2001)

5. Connor, J., Atlas, L.: Recurrent neural networks and time series prediction. In: Neural Networks, 1991., IJCNN-91-Seattle International Joint Conference on. vol. 1, pp. 301–306. IEEE (1991)
6. Consortium, A., et al.: Intelligent tyre systems-state of the art and potential technologies. APOLLO Deliverable D7 for Project IST-2001-34372. Also available at <http://www.vtt.fi/apollo/>, Technical Research Centre of Finland (VTT) (2003)
7. Dakhllallah, J., Glaser, S., Mammari, S., Sebsadji, Y.: Tire-road forces estimation using extended kalman filter and sideslip angle evaluation. In: American Control Conference, 2008. pp. 4597–4602. IEEE (2008)
8. De Jesús, O., Horn, J.M., Hagan, M.T.: Analysis of recurrent network training and suggestions for improvements. In: Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on. vol. 4, pp. 2632–2637. IEEE (2001)
9. Duchanoy, C.a.: Desarrollo de un modelo dinámico integral de un vehículo todo terreno con 6 subsistemas, su validación y estudio de maniobrabilidad y confort. Master's thesis, Centro de Investigación en Computación (2012)
10. Fischer, R.E., Tadic-Galeb, B., Yoder, P.R., Galeb, R.: Optical system design. Citeseer (2000)
11. Fortuna, L., Graziani, S., Rizzo, A., Xibilia, M.G.: Soft sensors for monitoring and control of industrial processes. Springer Science & Business Media (2007)
12. Ghaffari, A., Khodayari, A., Arvin, S., Alimardani, F.: An anfis design for prediction of future state of a vehicle in lane change behavior. In: Control System, Computing and Engineering (ICCSCE), 2011 IEEE International Conference on. pp. 156–161. IEEE (2011)
13. Hagan, M.T., Demuth, H.B., Beale, M.H., De Jesús, O.: Neural network design, vol. 20. PWS publishing company Boston (1996)
14. Hallonborg, U.: Super ellipse as tyre-ground contact area. Journal of Terramechanics 33(3), 125–132 (1996)
15. Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J.: Gradient flow in recurrent nets: the difficulty of learning long-term dependencies (2001)
16. Horn, J., De Jesús, O., Hagan, M.T.: Spurious valleys in the error surface of recurrent networks—analysis and avoidance. Neural Networks, IEEE Transactions on 20(4), 686–700 (2009)
17. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural networks 2(5), 359–366 (1989)
18. hpi.racing: Baja 5sc ss (2016), <http://www.hpiracing.com/en/kit/105734>
19. Ilonen, J., Kamarainen, J.K., Lampinen, J.: Differential evolution training algorithm for feed-forward neural networks. Neural Processing Letters 17(1), 93–105 (2003)
20. Keck, M.: A new approach of a piezoelectric vibration-based power generator to supply next generation tire sensor systems. In: Proceedings of IEEE Sensors. pp. 1299–1302 (2007)
21. Li, L., Wang, F.Y.: Advanced motion control and sensing for intelligent vehicles. Springer Science & Business Media (2007)
22. Makki, N., Pop-Iliev, R.: Piezoelectric power generation for sensor applications: design of a battery-less wireless tire pressure sensor. In: SPIE Microtechnologies. pp. 806618–806618. International Society for Optics and Photonics (2011)
23. Matsuzaki, R., Todoroki, A.: Passive wireless strain monitoring of actual tire using capacitance-resistance change and multiple spectral features. Sensors and Actuators, A: Physical 126(2), 277–286 (2006)
24. Matsuzaki, R., Todoroki, A.: Intelligent tires based on measurement of tire deformation. Journal of solid mechanics and Materials engineering 2(2), 269–280 (2008)

25. Matsuzaki, R., Todoroki, A.: Wireless monitoring of automobile tires for intelligent tires. *Sensors* 8(12), 8123–8138 (2008)
26. Miao, P., Liu, G., Zhang, D., Jiang, Y., Zhang, H., Zhou, H.: Sideslip angle soft-sensor based on neural network left inversion for multi-wheel independently driven electric vehicles. In: *Neural Networks (IJCNN), 2014 International Joint Conference on*. pp. 2171–2175. IEEE (2014)
27. Miller, S.L., Youngberg, B., Millie, A., Schweizer, P., Gerdes, J.C.: Calculating longitudinal wheel slip and tire parameters using gps velocity. In: *American Control Conference, 2001. Proceedings of the 2001*. vol. 3, pp. 1800–1805. IEEE (2001)
28. Phan, M.C., Beale, M.H., Hagan, M.T.: A procedure for training recurrent networks. In: *The 2013 International Joint Conference on Neural Networks (IJCNN)* (2013)
29. Slišković, D., Grbić, R., Hocenski, Ž.: Methods for plant data-based process modeling in soft-sensor development. *AUTOMATIKA: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije* 52(4), 306–318 (2012)
30. Suratgar, A.A., Tavakoli, M.B., Hoseinabadi, A.: Modified levenberg-marquardt method for neural networks training. *World Acad Sci Eng Technol* 6, 46–48 (2005)
31. Toplar, H.: Experimental analysis of smart tires (2014)
32. Xia, K., Yang, Y.: Three-dimensional finite element modeling of tire/ground interaction. *International journal for numerical and analytical methods in geomechanics* 36(4), 498–516 (2012)
33. Zhang, W., Ding, N., Yu, G., Zhou, W.: Virtual sensors design in vehicle sideslip angle and velocity of the centre of gravity estimation. In: *Electronic Measurement & Instruments, 2009. ICEMI'09. 9th International Conference on*. pp. 3–652. IEEE (2009)