

Universidad Internacional de La Rioja (UNIR)

ESIT

Máster Universitario en Inteligencia Artificial

AI4AO: The inevitable future to see the stars

Trabajo Fin de Máster

Presentado por: Ramírez Trujillo, María de la Cruz

Director/a: Fernández Isabel, Alberto

Co-Director/a: López López, Roberto

Ciudad: Cartagena (Murcia)

Fecha:

Agradecimientos

A Stephen John Weddell, por su inmenso apoyo en este proyecto. Gracias por haberme apoyado con sus conocimientos tanto en Inteligencia Artificial como en Óptica Adaptativa.

A Esther Soria Hernández, gracias a la cual se ha obtenido acceso a la posibilidad de simular el banco óptico de Óptica Adaptativa como librería en Python. Su librería es de increíble valor, y no puedo ofrecerle los suficientes reconocimientos.

A Roberto López López, por sus conocimientos y consejos tan valorados para el correcto desarrollo científico de este trabajo final de máster. Su gran experiencia en el sector ha sido de increíble ayuda, y no puedo estar lo suficientemente agradecida.

Y a todas las personas que me han apoyado durante este duro proceso. A todos vosotros.

Resumen

Se presenta una metodología basada en Aprendizaje Automático para solventar el problema de uso de controladores para la corrección en lazo cerrado en técnicas de Óptica Adaptativa. Se estudian varios modelos para discutir la eficacia de distintas metodologías frente al problema presentado. El modelo final, el presentado en este Trabajo Final de Máster como RNN+LSTM, tiene la capacidad de modelar un frente de onda en todo el rango de nodos de Zernike en menos de un milisegundo por matriz, demostrando así su eficacia para eliminar el uso de otros tipos de control más tradicionales. Posee mayor precisión en cálculo a la hora de predecir la evolución de una serie temporal de un frente de onda, así como mayor optimización de uso de procesador y tiempos de evaluación. Por último, se discute una serie de mejoras para trabajos futuros, donde se podría discutir la eliminación total o parcial de sensores y solo utilizar tecnologías basadas en Inteligencia Artificial.

Palabras Clave: Óptica Adaptativa, Aprendizaje Automático, Control Adaptativo

Abstract

A novel, Machine Learning method is presented to solve control-specific problems related to Adaptive Optics techniques, as well as an evidence of its use against other well-known techniques of adaptive control. Several methods are discussed for solving the presented problem [aberration correction in real time], from which the use of LSTM+RNN is highly recommended and defended in this report. The LSTM+RNN model has the capacity of calculating not just the first Zernike nodes but the entire node range, at an optimal calculus speed of less than one millisecond per matrix, demonstrating its potency against more traditional opto-mechanical controllers. Plus, the LSTM+RNN model has showed improved results over time, overall improved processor performance and optimized calculus of a wavefront, Zernike based model. Furthermore, several improvements are discussed at the end of the report, where Artificial Intelligence methods are the key to new, improved Adaptive Optics.

Keywords: Adaptive Optics, Machine Learning, Adaptative Control

Índice de contenidos

1. Introducción	1
1.1 Motivación.....	6
1.2 Planteamiento del trabajo	7
1.3 Estructura de la memoria.....	9
2 Contexto y estado del arte	11
2.1 Estado del Arte en Óptica Adaptativa.....	12
2.2 Redes Neuronales Recurrentes	12
2.2.1 LSTM	14
2.3 Conclusiones del estado del arte.....	15
3. Objetivos y metodología de trabajo	17
3.1 Objetivo general.....	17
3.2 Objetivos específicos.....	17
3.3 Metodología del trabajo	19
4. Identificación de requisitos.....	23
4.1 Entorno de desarrollo	23
4.2 Datos	23
4.3 Modelado de la LSTM-RNN.....	24
4.4 Evaluación de los resultados	25

5. Descripción de la metodología.....	27
5.1 Simulación de los datos.....	27
5.2 Estudio de los datos generados de la simulación.....	28
5.3 Diseño y entrenamiento.....	32
5.3.1 Preparación previa de los datos	34
5.3.2 Entrenamiento	37
6. Evaluación	40
6.1 Modelo LSTM	40
6.2 Comparación con otros modelos	43
7. Conclusiones y trabajo futuro	49
7.1 Conclusiones	49
7.2 Líneas de trabajo futuro.....	50
8. Bibliografía	51
Anexos	55
Anexo I. LSTM para AO: resultados.....	55
Anexo II. Artículo de investigación.....	1

Índice de tablas

Tabla 1. Polinomios de Zernike fundamentales.....	5
Tabla 2. Medidas del error rms introducido en el sistema optomecánico del proyecto LGS AO	6
Tabla 3.-Valoración modelo LSTM	43
Tabla 4. Valoración modelo LSTM mas PID.....	44
Tabla 5.-Resultados del modelo ANN	45

Índice de figuras

Figura 1. Efecto de la aplicación de Óptica Adaptativa en el telescopio de 3.6 m del observatorio La Silla, en el desierto de Atacama (Chile). Fuente: European Southern Observatory.	2
Figura 2. Sistema Optomecánico de Óptica Adaptativa. Fuente: Andor Technology, Oxford Instruments.	2
Figura 3 . Representación gráfica de los 25 primeros polinomios de Zernike.....	4
Figura 4. Imagen recibida de un sensor Shack-Hartmann. Fuente: Proyecto AOLI.....	7
Figura 5. Estructura interna de una RNN. Fuente: Stanford University, CS 230 – Deep Learning	13
Figura 6. Izquierda: Banco de Óptica Adaptativa utilizado en AOLI. Derecha: Corrector utilizado en AOLI. Fuente: Proyecto AOLI [5] [9].....	18
Figura 7. Flujograma de la metodología de trabajo. Fuente: autor.....	20
Figura 8. Lazo de control para AO basado en LSTM-RNN. Fuente: Autor	26
Figura 9. Definición optomecánica de la fuente de luz y del telescopio. Fuente: Autor / ALIOLI [5].....	26
Figura 10. Extracción de datos de la estrella de referencia Fuente: Autor / ALIOLI [5].....	26
Figura 11. Extracción de datos de la estrella de referencia Fuente: Autor	26
Figura 12. Simulación de los datos. Fuente: Autor / ALIOLI[5].....	27
Figura 13. Arriba: conversión de los datos obtenidos en la simulación a dataframes. Abajo: dataframe generado como dato de entrada. Fuente: Autor.....	29
Figura 14. Dataset de la secuencia temporal. Fuente: Autor	30
Figura 15. Transformación de la secuencia temporal. Fuente: Autor	30
Figura 16. Aberración con respecto a un píxel i,j dentro de una secuencia temporal. Fuente: Autor	30
Figura 17. Dataset de la representación del frente de onda para cada imagen i respecto a los siete primeros nodos de Zernike. Fuente: Autor.....	31
Figura 18. Estudio estadístico del dataset de entrada. Fuente: Autor	31
Figura 19.-Modelo LSTM-RNN aplicado a AO. Fuente: Autor.....	32
Figura 20. Modelo ANN simplificado para AO. Fuente: Autor	33

Figura 21. GridSearch en el modelo simplificado ANN. Fuente: Autor.....	33
Figura 22. Split del conjunto de datos para entrenamiento y validación de los modelos propuestos. Fuente: Autor	34
Figura 23. Normalización del conjunto de datos. Fuente: Autor.....	35
Figura 24. Redimensionalización del conjunto de datos. Fuente: Autor.....	35
Figura 25. Modelado del conjunto de datos como un control PID. Fuente: Autor	36
Figura 26. Compilación del modelo LSTM. Fuente: Autor.....	37
Figura 27. Training del modelo. Fuente: Autor	37
Figura 28. Curva de pérdidas por epoch del modelo LSTM. Fuente: Autor	40
Figura 29. Precisión sobre la validación. Fuente: Autor	41
Figura 30. Resultado temporal modelo LSTM. Fuente: Autor	42
Figura 31. Valoración del error en el modelo LSTM. Fuente: Autor	42
Figura 32. Comparación entre los valores calculados y verdaderos para la aberración tipo tilt. Fuente: Autor	42
Figura 33. Gráfica de función de coste para LSTM+PID. Fuente: Autor	44
Figura 34. Gráfica de precisión para LSTM PID. Fuente: Autor	44
Figura 35. Curva de pérdidas del modelo ANN	45
Figura 36. Comparativa de costes entre modelos. Fuente: Autor	46

1. Introducción

La astrofísica, un área de la ciencia dónde se lleva a cabo la innovación más puntera de la tecnología, junto con la de uso militar y médico, se está apoyando cada vez más desde las últimas décadas en las metodologías de la Inteligencia Artificial, impulsando su desarrollo. La pregunta puede radicar en, ¿Dónde se aplica la Inteligencia Artificial en esta ciencia?

Como es conocido, los telescopios son cada vez de mayor tamaño para poder hacer frente a dos problemas fundamentales: la resolución y el alcance. Esto se puede entender de una forma más simplificada: a mayor diámetro de telescopio, más luz permite recoger, y esto se convierte en mayor poder en la observación. Por supuesto, la nitidez va de la mano con este concepto, y es donde entra en juego el término de la resolución; cuanto mayor es la apertura del instrumento, mayor poder resolutivo alcanza. Esto es muy beneficioso, ya que permite distinguir estrellas con menor ángulo de separación (Friedlander, M.W) [1].

Aun con el avance tecnológico tan grande que estamos experimentando hoy en día, hay características tecnológicas y físicas que impiden que logremos las capacidades de un observatorio espacial en la tierra. Pongamos como caso el Telescopio Espacial Hubble, de 2.5 metros de apertura. Pese a que su tamaño se considera de telescopio de clase “pequeña”, su resolución angular es ocho veces mayor a la que se presenta en telescopios terrestres de su misma clase (Mackay, C.) [2].

Se pueden alcanzar, sin embargo, prestaciones de resolución angular comparables (aunque no mejores) al HST en telescopios terrestres con el uso de la **Óptica Adaptativa**. La definición dada por el European Southern Observatory [3] para esta técnica es la siguiente:

“Método óptico que implica el uso de espejos deformables, controlados por ordenadores que pueden corregir en tiempo real las perturbaciones en la imagen provocadas por la atmósfera de la tierra”

Como se puede apreciar en la figura [1], el efecto es notablemente beneficioso a la hora de realizar observaciones astronómicas. Lo que parecía una sola estrella, es en realidad una estrella binaria.

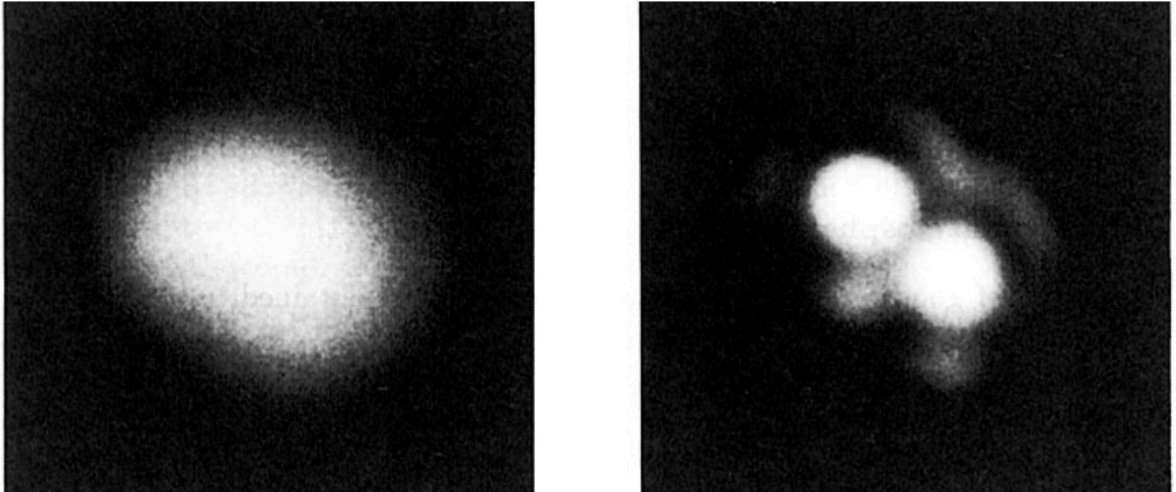


Figura 1. Efecto de la aplicación de Óptica Adaptativa en el telescopio de 3.6 m del observatorio La Silla, en el desierto de Atacama (Chile). Fuente: European Southern Observatory.

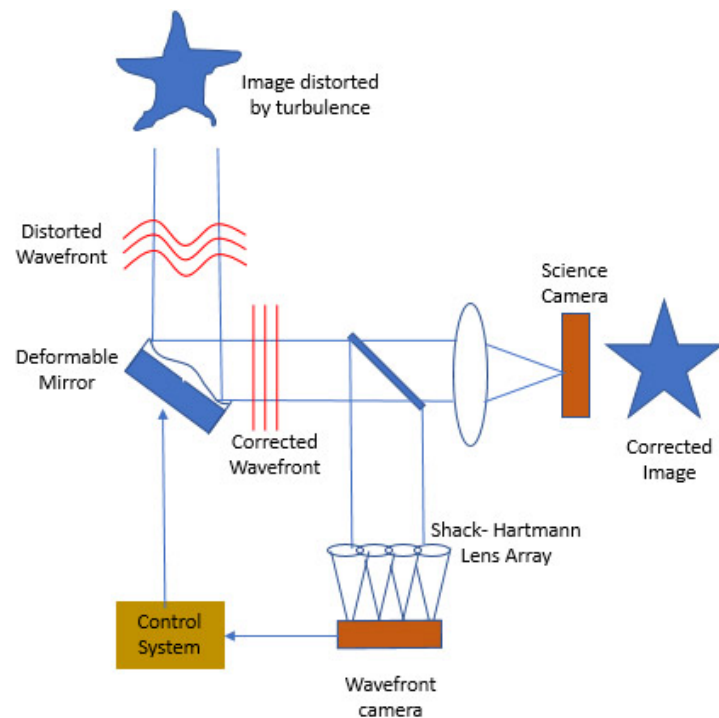


Figura 2. Sistema Optomecánico de Óptica Adaptativa. Fuente: Andor Technology, Oxford Instruments.

Crear este sistema corrector desde el punto de vista de la instrumentación óptica no es tarea fácil. Como se puede apreciar en la figura [2], se requieren de diversos sensores que, coordinándose sincronamente, pueden dar lugar a tal mejora.

El proceso empieza cuando el telescopio enfoca un objetivo, como pudiera ser una estrella como en el caso de la figura [1]. El espejo deformable, dejaría pasar en una primera iteración el frente de onda sin aplanar directamente al splitter. El splitter es el elemento óptico capaz de dividir un haz de luz en dos caminos distintos. Uno de los caminos lleva a la cámara de ciencia, mientras que el otro es dirigido al lazo cerrado del sistema corrector.

El sensor óptico pudiera ser de tres tipos:

- Piramidal
- Geométrico
- Shack-Hartmann

Lo que diferencia dichos sensores, algorítmicamente hablando, es en el proceso de análisis que requieren para extraer la forma de onda en función de modos de Zernike. Se necesita descomponer la onda en sus nodos fundamentales para lograr comprender cómo se ha deformado esta al atravesar la atmósfera.

Para entender mejor estos conceptos, se debe conocer primero matemáticamente qué es un Frente de Onda y qué es un Polinomio de Zernike, con sus derivados modos. En Óptica Adaptativa (Leigui de Oliveira, M.A) [4], se describe un Frente de Onda como la cantidad que es alterada para compensar el desfase introducido por las turbulencias. Matemáticamente, se describe como:

$$\Psi = A(x, y)e^{i\phi(x, y)} \quad (1)$$

Donde $A(x, y)$ es la función que describe la apertura del telescopio y $\phi(x, y)$ es la función que describe la fase.

Los polinomios de Zernike son una serie de polinomios ortogonales que provienen de la expansión de la función de un frente de onda para sistemas ópticos con pupilas circulares. Estos polinomios pueden ser pares o impares, descritos de la siguiente forma:

- Polinomios pares: $Z_n^m(\rho, \theta) = R_n^m(\rho) \cos(m\theta)$ (2)

- Polinomios impares: $Z_n^{-m}(\rho, \theta) = R_n^m(\rho) \sin(m\theta)$ (3)

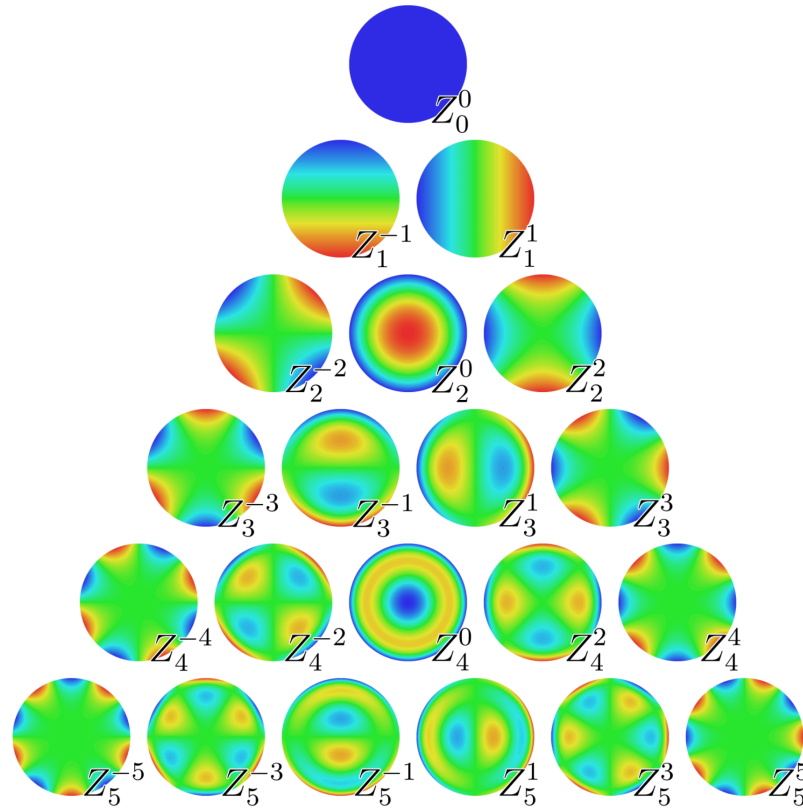


Figura 3 . Representación gráfica de los 25 primeros polinomios de Zernike

Donde m y n son índices no negativos, con la condición de que $n > m$, θ es el ángulo azimutal ($0 \leq \theta \leq 2\pi$) y ρ es la distancia radial dentro del disco unidad. $R_n^m(\rho)$ pertenece al cálculo del polinomio radial, tal que:

$$R_n^m(\rho) = \sum_{k=0}^{\frac{n-m}{2}} \left[\frac{(-1)^k (n-k)!}{k! \left(\frac{n+m}{2} - k\right)! \left(\frac{n-m}{2} - k\right)!} \right] \rho^{n-2k} \text{ para } n-m \text{ par} \quad (4)$$

$$R_n^m(\rho) = 0 \text{ para } n-m \text{ impar} \quad (5)$$

Esto es fundamental conocerlo al detalle para este trabajo, ya que como se explicará más adelante, la corrección del sistema de Óptica Adaptativa se basa en obtener los treinta nodos de Zernike para corregir el frente de onda.

Tabla 1. Polinomios de Zernike fundamentales

Polinomio	Tipo de aberración
Z_0^0	Pistón
Z_1^{-1}	Inclinación vertical
Z_1^1	Inclinación horizontal
Z_2^{-2}	Astigmatismo
Z_2^0	Desenfoque

En la tabla 1 se describirá el tipo de aberración óptica que representan los cinco primeros polinomios de Zernike. Los posteriores representan aberraciones que provienen de los anteriores.

El sistema optomecánico descrito en la figura 2 obtendrá una descripción del frente de onda en función de dichos polinomios. Gracias a esto, se puede representar gráficamente dicha onda y corregirla, a través de un elemento optomecánico perteneciente a la corrección llamado Espejo Deformable.

Un Espejo Deformable no es sino una lámina de cristal reflectante que se deforma. La forma en la que se deforma es la diferenciación entre varios tipos de Espejos Deformables, pero se puede reducir en actuadores que deforman dicha lámina para contrarrestar el frente de onda deformarlo y aplanarlo. Finalmente, el haz de luz corregido se llevaría a la cámara de ciencia.

Este Trabajo Final de Máster forma parte de un proyecto del Instituto de Astrofísica de Canarias. Dicha institución ha cedido amablemente la utilización de su banco de ensayos para Óptica Adaptativa en laboratorio, proporcionado uno de los sensores fundamentales para este trabajo [5], formando parte del proyecto ALIOLI, anteriormente conocido como AOLI. Por otro lado, se cuenta con la colaboración del equipo CDA [6] de la Universidad de Canterbury (Christchurch, Nueva Zelanda), que proporcionan otro banco de ensayos especializado en otros sensores distintos a los utilizados por los del Instituto de Astrofísica de Canarias.

Tabla 2. Medidas del error rms introducido en el sistema optomecánico del proyecto LGS AO

Wavefront Error Term	nm rms
Atmospheric Fitting	122
Telescope Fitting	66
Science Camera	110
Wavefront Bandwidth	182
Wavefront Measurement	216
LGS Focus Error	70
Focus Anisoplanatism	181
LGS High Order Error	80
Calibration Errors	30
Miscellaneous	73
Total High Order Wavefront Error	402
Tip-tilt Bandwidth	247
Tip-tilt Measurement	214
Tip-tilt Anisoplanatism	115
Total Tip-tilt Wavefront Error	346
Total Wavefront Error	531

1.1 Motivación

Para entender la motivación de este trabajo es necesario entrar en detalles con el funcionamiento del equipo de Óptica Adaptativa. Como ilustra (Wizinowich) [7], el enemigo fundamental para el éxito del sistema optomecánico básico de Óptica Adaptativa es el tiempo.

Las aberraciones atmosféricas producen alteraciones de visibilidad en una frecuencia del orden de 10 milisegundos [8]. Con esta milésima franja temporal, deductivamente se pudiera pensar en cómo se logra la corrección en tiempo real en los observatorios. Por otro lado, hay que contar no solo con el retraso temporal que se obtiene por el empleo de múltiples sensores, sino también el error de medida que estos introducen. La tabla 2 refleja el error que introducen las diferentes partes de la instrumentación para el caso de [7], perteneciente al proyecto LGS AO, del Keck Observatory (Hawaii).

Y si lo descrito anteriormente no pareciera suficiente, hay que añadir el error de cálculo que introduce los algoritmos correctores dentro del sistema de control. Se puede exponer como caso como el sistema de control actual del proyecto AOLI, reflejado en (Colodro, C.) [9] tarda unos milisegundos en realizar los cálculos para poder corregir y adaptar el sistema optomecánico. Esto no es realmente eficiente, en cuestiones de tiempo para la variación de las aberraciones.

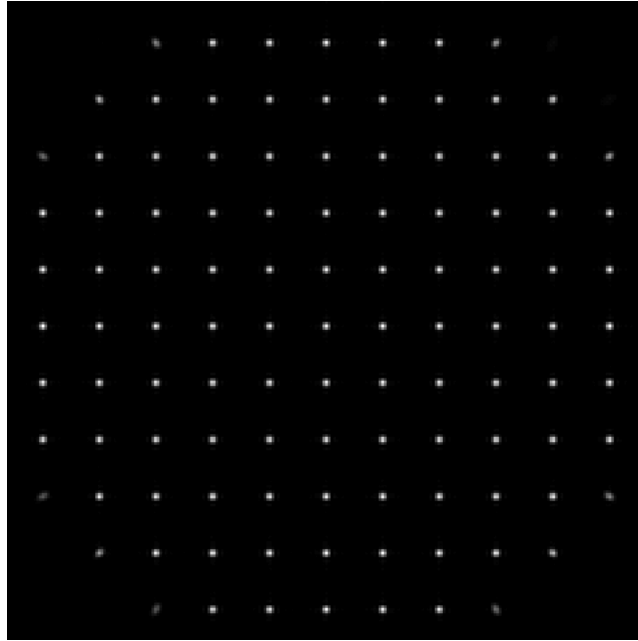


Figura 4. Imagen recibida de un sensor Shack-Hartmann. Fuente: Proyecto AOLI

Este problema se podría ver increíblemente reducido si se aplicase un sistema correctivo basado en Inteligencia Artificial, el cual será la ocupación de este Trabajo Final de Máster. Haciendo un sistema basado en Redes Neuronales, capaces de predecir la forma de onda a corregir esperada directamente de la imagen de ciencia, se lograría reducir mucho tiempo de procesado, eliminando incluso algunos procesos opto-mecánicos que intervienen en la adquisición.

1.2 Planteamiento del trabajo

El problema anteriormente descrito requiere del diseño de un Sistema Cognitivo, capaz de trabajar en alto rendimiento y con series de datos dependientes del tiempo. Debe de ser eficiente, muy preciso y rápido. Este sistema deberá proporcionar la forma de frente de onda en función de polinomios de Zernike o una matriz de desplazamientos, para que así se pueda corregir en tiempo real y saber al instante qué elementos intervienen en la aberración.

Un sensor típico para leer un sensor de frente de onda es un Shack-Hartmann. Este sensor recibe un haz de luz y devuelve lo observado en la figura 4.

Leyéndose una imagen de referencia, de un caso “ideal” sin perturbación, se estudia la desviación de dichos puntos de la imagen obtenida.

El método tradicional de corregir un espejo deformable en función de la desviación de los centroides observados en la figura 4 es calculando los polinomios de Zernike y, calculando la matriz inversa de desplazamientos, corregir la posición de los actuadores en el espejo.

Otro sensor con el que se deberá trabajar en este proyecto es el conocido como TP3. Según lo enunciado por (Dam, V.) [10], se trata del primer sensor que implementa físicamente la reconstrucción geométrica de un frente de onda. Se diferencia, de un Shack-Hartmann, en que para calcular los polinomios de Zernike este se alimenta de la imagen de dos pupilas distintas: una extrafocal y una intrafocal.

En definitiva, ambos sensores producen la misma salida, ya sea en modos de Zernike o en cálculo de matriz de desviaciones. Lo que realmente diferencia estos sensores es la entrada de información, y como la procesan. La parte de procesado de la información es el problema que se intenta remediar mediante redes neuronales, ya que son de gran coste computacional. En estos procesos no solamente intervienen cálculo matricial de alta densidad, sino que se incluye el uso de controladores tipo Proporcional-Integrador-Derivativo de la ingeniería clásica.

Se necesitará un sistema multi-salida, basado en regresión. La metodología propuesta principal para resolver este problema es añadir un sistema basado en Redes Neuronales Recurrentes. Como se menciona en [11], son un tipo de redes neuronales especializadas en trabajar con requisitos temporales, como es en el Procesado del Lenguaje natural. Lo que las distingue de otros sistemas cognitivos es su capacidad de “memoria”, añadiendo la capacidad de influenciar la salida respecto a datos anteriores. Es en estos casos donde se asume dependencia de los datos de entrada con los de salida. Se necesitará hacer que el sistema multi-salida ofrezca tanto los nodos más esenciales de Zernike como una predicción de la matriz de deltas, que directamente corrija el sensor.

Esto podría ser de increíble utilidad, ya siendo la atmósfera variante, un sistema con capacidad de adaptarse a nuevas condiciones a la vez que es capaz de recordar anteriores podría ser de increíble beneficio para un sistema de Óptica Adaptativa. Otra mejora que implementa el uso de redes neuronales es que independiza el algoritmo del sensor de entrada. Como se mencionaba anteriormente, cada sensor necesitaba un procesado distinto para poder controlar el espejo deformable. En este trabajo se intentará independizar la red neuronal de los sensores de entrada.

Como último detalle, se estudiará el uso de un sensor Shack-hartmann dentro del lazo de corrección y memoria de la Red Neuronal Recurrente, potenciando la sensibilidad y la precisión cuando el sistema utiliza un TP3. Se plantea este objetivo debido a que un sensor Shack-Hartmann es mucho menos preciso que un TP3.

1.3 Estructura de la memoria

Una vez contemplado una introducción al problema presentado de óptica adaptativa, tanto sus principios físicos como la problemática, los siguientes apartados constarán de lo siguiente:

- Contexto y estado del arte. Aquí se abordará el estado y trabajos actuales de la Inteligencia Artificial aplicados a la Óptica Adaptativa.
- Objetivos y metodología de trabajo. Se profundizará en la metodología diseñada para resolver el problema descrito en la introducción.
- Contemplación de los objetivos necesarios para el correcto funcionamiento de la metodología desarrollada
- Descripción en profundidad de la metodología, así como análisis de los experimentos llevados a cabo para corroborar los resultados
- Evaluación de los resultados, con comprobación en laboratorios reales.
- Conclusiones y líneas de trabajo futuro para un posible estudio doctoral.

2 Contexto y estado del arte

El lugar adecuado para empezar a buscar sobre el estado del arte de este ámbito de la ciencia no es más sino que el SPIE (del Inglés “*Society of Photographic Instrumentation Engineers*”) [12], sociedad sin ánimo de lucro especializada en tecnología de óptica. Aquí se encuentran los estudios más novedosos en el ámbito de la óptica, y en especial, de la Óptica Adaptativa.

En el sistema descrito por (Escárte, P.) [13], se utiliza un sistema basado en Redes Neuronales Artificiales para simular el comportamiento de un sensor de frente de onda, en substitución de los descritos en la introducción. Sin embargo, utilizan un sistema de control independiente. No enuncian resultados de mejora en cuestiones de trabajo en tiempo real, pero sí verifican la precisión a la hora de predecir la forma de un frente de onda.

Otro sistema es el ofrecido por (Landman. R.) [14], donde se diseña un sistema auto-optimizador para Óptica Adaptativa basado en Reinforcement-Learning. Se aplica sobre una Red Neuronal Recurrente como controlador del sistema de Óptica Adaptativa, pero especializado en el control de las aberraciones tipo Tip-Tilt. El propuesto en este trabajo se globaliza a varios tipos de sensores de entrada y a todos los modos de zernike, abarcando un paso más allá al mencionado en este trabajo.

También se ha utilizado Deep Neural Networks para eliminar el ruido presente en la adquisición de imágenes astronómicas en procesos de Óptica Adaptativa, como se presenta en (Pou, B.) [15]. En (Thomas, D.) [16], se emplea para aumentar la calidad óptica de la imagen durante el proceso de Óptica Adaptativa Redes Convolucionales Recurrentes. En el camino también de la corrección de imagen para Óptica Adaptativa nos encontramos con (Quesnel, M.) [17], mostrando cómo la implementación de dos redes neuronales convolucionales para estimar los errores de frente de onda proveniente de simulaciones de Point-Spread-functions en distintos regímenes de aberración, pero no muestran resultados reales.

Todos los anteriores trabajos fueron publicados en dos mil veinte, por lo que son bastante recientes en este campo, mostrando la tecnología más actual en el ámbito de la Óptica Adaptativa junto con el de Inteligencia Artificial. Lo que todos tienen en común, aunque muchos se queden solo con datos simulados, es que llegan a la conclusión del gran potencial que tiene el Aprendizaje Automático en este ámbito de la ciencia.

Como ha podido observarse, muchos de los trabajos presentados anteriormente reemplazan directamente un sensor, o se dedican a la parte de control, o solamente a mejorar la calidad de imagen. En este Trabajo Final de Máster se propondrá el estudio de una metodología que

abarcará corrección de imagen, optimización de cálculo y control, con la ventaja de tener la oportunidad de no solo poder simular los datos en ordenador, sino en dos laboratorios de óptica reales y profesionales, como los proporcionados por el Instituto de Astrofísica de Canarias y la Universidad de Canterbury.

2.1 Estado del Arte en Óptica Adaptativa

En cuanto al campo de la Óptica Adaptativa, cabe mencionar los progresos que se están llevando a cabo. En Soria, E. [25], se presenta un nuevo concepto de sistema opto-mecánico para el proyecto AOLI donde se introduce un nuevo diseño para este propósito. La novedad de este instrumento es que está basado en la modularización de sus componentes: desde el simulador, el corrector de frente de onda, hasta los sensores que intervienen en el proceso. Esto hace que la Óptica Adaptativa pueda aplicarse también en telescopios de tamaño mediano y pequeños, ya que normalmente (y debido a sus costes), solo era implementada en telescopios de gran tamaño.

En Taghinia, P. [26], se implementa una metodología que recurre a la eliminación de sensores de frente de onda para la corrección de Tip/Tilt. En este caso, se utilizan métodos matemáticos para eliminar estos dos modos de Zernike, y debate su eficacia en términos de precisión y rapidez. Este caso infiere que no es una idea radical el reducir componentes opto-mecánicos de la mesa de Óptica Adaptativa, como se pretende en este trabajo final de máster, sino que es una medida deseable para la mejora de esta metodología.

Por otro lado, en Liu, J. [27] se presenta un controlador opto-mecánico diseñado para corregir los modos de Zernike Tip/Tilt en tiempo real. Los resultados presentados en este artículo corroboran en que la simplificación de la obtención de órdenes menores de zernike ayudan notoriamente en la corrección de la perturbación atmosférica, reduciendo costes de computación y tiempos de corrección.

2.2 Redes Neuronales Recurrentes

Hoy en día podemos ver una diversa explosión de artículos de numerosas aplicaciones de las redes neuronales recurrentes, tanto en el mundo científico como en el industrial. No solamente son dedicadas a las técnicas del Procesado del Lenguaje Natural, sino podemos encontrar aplicaciones en todo de tipo de campos.

Este tipo de redes son especialmente investigadas para entornos donde sea primordial trabajar en tiempo real, con gran velocidad de cómputo y de adaptación a secuencias que dependen del tiempo. Según Amidi, A.[18], son un tipo de redes neuronales que permite que anteriores estados sean usados como realimentación del sistema. Su estructura interna se puede ver reflejada en la figura [5].

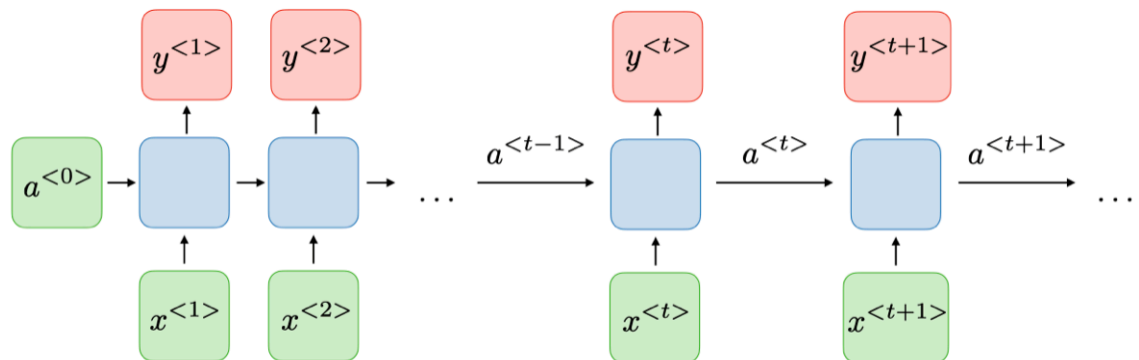


Figura 5. Estructura interna de una RNN. Fuente: Stanford University, CS 230 – Deep Learning

Las RNN también añaden versatilidad al sistema: son capaces de adaptarse a múltiples entradas y generar diversas salidas. Por lo general, se trabajan con tres tipos de estructuras:

- Una sola entrada que genera una secuencia en la salida
- Una secuencia como entrada que activa una única salida
- Una secuencia de datos de entrada que genera una secuencia de salida.

En la figura [5] se puede observar que la salida no depende directamente de la entrada y la función de activación, sino que pasa por un “*hidden state*” o estado interno. Es la formulación de este estado el que va a diferenciar entre varias clases de redes neuronales recurrentes.

Este estado interno es comúnmente formulado de la siguiente forma:

$$h_t = \tanh (W_{xh} X_t + W_{hh} h_{t-1}) \quad (6)$$

Donde:

- W_{xh} y W_{hh} son matrices de pesos o parámetros que modelan la red neuronal
- h_{t-1} es el estado interno anterior y h_t es el nuevo estado interno

- X_t es la entrada al sistema para un instante t
- $Tanh$ es la función hiperbólica

A la fórmula (6) es la que se conoce como “Vanilla” RNN. Se verá a continuación otras vertientes actuales para la definición de esta función.

Existen diversas maneras de entrenar las RNN. Según documenta Raj, J.S. [19], las dificultades comúnmente encontradas al intentar realizar regresión para datos no lineales se pueden solventar utilizando RNN con algoritmos de *Support Vector Machine* en lugar de *Back-Propagation*.

2.2.1 LSTM

Rothstein M.E. [20] desarrolla un artículo sobre el funcionamiento más actual de los LSTM-RNN en 2019. Una RNN estándar no puede puentear más de 5 a 10 instantes de tiempo, debido a que el error introducido de las señales provenientes del back-propagation tienden a crecer o decrecer con cada intervalo temporal. Esto puede causar un grave problema; si los gradientes del back-propagation incrementa a valores desmesurados, ocurriendo el llamado “*exploding gradient*”, o por el contrario que se anulen (*vanishing gradients*).

Las arquitecturas LSTM están diseñadas para solventar el problema temporal para instantes de tiempo lejanos, es decir, solventan el problema de memoria que encontramos en las RNN “vanilla”. Además, no sufren de la problemática mencionada con respecto a los gradientes, por lo que representan una gran mejora respecto a la arquitectura vanilla.

En Xuewen, L. [21] (2019) se hace referencia a un sistema basado en ANN-LSTM para afrontar el problema temporal de la Óptica Adaptativa, basándose en un control de lazo abierto y un modelo para datos no lineales. Esto se aproxima más al objetivo de este trabajo final de grado, aunque carezca de realimentación para la corrección. Su objetivo principal es predecir el frente de onda inmediato basándose en secuencias temporales. Consiguieron una reducción del error del 70% sobre datos simulados. Para mejorar la velocidad de computación, usaron ajustes de los pesos del LSTM a modo de optimización.

Finalmente, Chen Y. [22] defiende un sistema de predicción de frente de ondas con nodos de Zernike basado en Redes Neuronales Recurrentes. Este artículo da un paso más al propuesto por [21], ya que se centran en la base de control a posteriori de la obtención del frente de onda. El cálculo de los nodos de Zernike lo realizan independientemente de la RNN, y lo que sí se objetiviza el modelo presentado es para el cálculo de la matriz de influencia para el

control del espejo deformable. Se hace referencia a la matriz de influencia como la matriz que convierte de nodos de zernike a valores de voltaje para el control del espejo deformable.

2.3 Conclusiones del estado del arte

Como se ha comprobado en los puntos explicados anteriormente, existen numerosas posibilidades para implementar un controlador predictivo basado en Inteligencia Artificial para Óptica Adaptativa.

Dependiendo de la finalidad, podemos encontrarnos controladores basados en imagen y en redes convolucionales, otros dedicados exclusivamente a la optimización de cálculo... pero en común concluyen que las metodologías ofrecidas por el aprendizaje automático y, en especial los sistemas cognitivos, ofrecen amplias ventajas respecto a los sistemas tradicionales de control y tratamiento de datos.

3. Objetivos y metodología de trabajo

En el siguiente apartado se explicará los objetivos a alcanzar por este Trabajo Final de Máster.

3.1 Objetivo general

El objetivo de este trabajo final de máster consistirá en la obtención de un modelo basado en redes neuronales que pueda sustituir a un controlador convencional utilizado en un sistema opto-mecánico tradicional de Óptica Adaptativa, mejorando sus prestaciones.

Gracias a la Inteligencia Artificial se puede reducir notablemente los recursos empleados en proyectos de ingeniería, sustituyendo diversos elementos (tanto electrónicos como mecánicos) por algoritmos. Los beneficios de esta acción, y los cuales forman parte del objetivo de este trabajo, serán los siguientes:

- Añadir mayor precisión
- Reducir tiempos de cómputo
- Automatizar el proceso

Por ejemplo, al sustituir una cadena de sensores por un solo algoritmo de control, se reducen características como errores de medida, ruido, etc. En el caso que ocupa este trabajo, al reemplazar el controlador convencional no solo se mejora la precisión, sino que se aumenta la rapidez del proceso ya que se requiere de menos elementos.

Se pretende, por lo tanto, diseñar una metodología actualizada basada en redes neuronales que pueda reducir los componentes usados en el banco opto-mecánico de AOLI [5], a la vez que aumente la precisión de corrección del frente de onda.

3.2 Objetivos específicos

La metodología presentada en este trabajo deberá hacer frente a una serie de requisitos específicos para el proyecto donde se aplica. En este caso, nos encontramos con el sistema presentado en la figura 6.

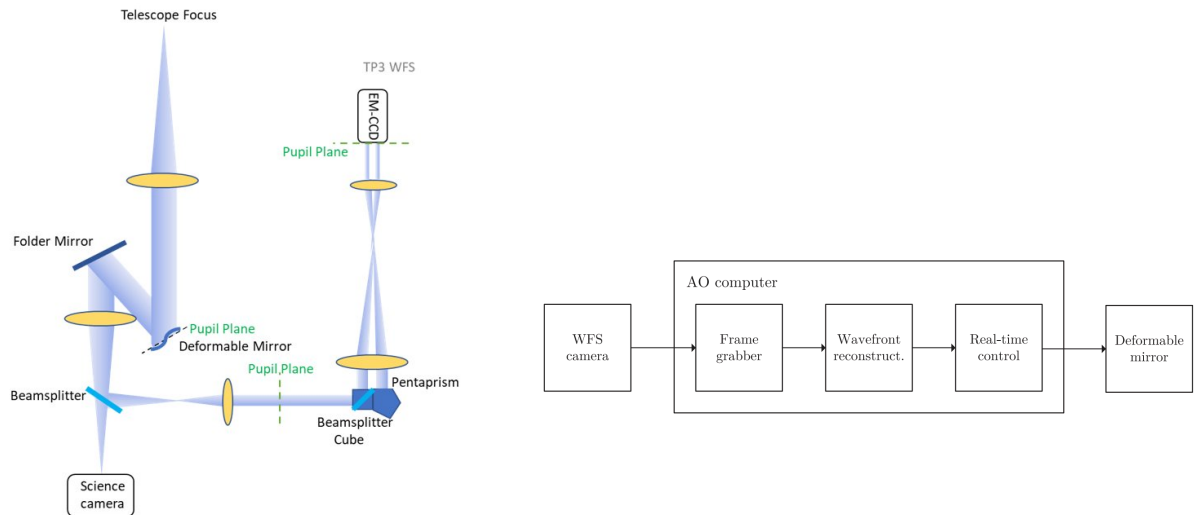


Figura 6. Izquierda: Banco de Óptica Adaptativa utilizado en AOLI. Derecha: Corrector utilizado en AOLI. Fuente: Proyecto AOLI [5] [9]

En este escenario, se debe adaptar una metodología que sea capaz de trabajar con la matriz de puntos de desplazamiento sobre una imagen de referencia para obtener un frente de ondas modelado, para que se pueda controlar posteriormente un espejo deformable. El control de dicho espejo deformable se consigue multiplicando la matriz de nodos de zernike por la matriz de caracterización del espejo deformable.

La matriz de caracterización del espejo deformable se obtiene evaluando la variación de voltaje que se produce para cada actuador dado un frente de onda en base a modos de zernike. Como esto entra dentro del modelado del propio espejo deformable, no será necesario llegar a introducir este cálculo dentro de la metodología, ya que es una multiplicación matricial a los resultados obtenidos por la propia red neuronal.

Por otro lado, la metodología final debe presentar la mayor simplicidad posible. Dado a que se quiere reducir tiempos de costes, se puede permitir cierto error en cuanto a precisión (aunque no muy elevado), si eso conlleva a mayor velocidad de procesamiento.

Para facilitar el proceso, se reduce el área de la imagen al área o Región de Interés (conocido como ROI). Esto simplifica la matriz a trabajar, ya que reduce de un tamaño de 256x256 píxeles por imagen (como se verá en capítulos posteriores) a la evaluación de 178 puntos. La red neuronal debe ser capaz de trabajar con este tamaño de datos de entrada, a la vez que debe generar una salida adaptada a los nodos de zernike necesarios.

Como mínimo, debe de ser capaz de calcular los 7 primeros nodos de Zernike. Sin embargo, se evaluará el comportamiento de la red neuronal frente al cálculo de los 23 nodos totales de

Zernike. Se considerará de gran éxito el trabajo si se logra obtener un modelo que logre obtener los nodos en su totalidad, aunque no es primordial. Se tomará la decisión, como se verá en capítulos posteriores, dependiendo del coste computacional que esto conlleve.

En resumen, se debe obtener una metodología que trabaje con una secuencia de entrada y con una secuencia de salida, capaz de adaptarse a los cambios atmosféricos a la vez que se obtiene una precisión elevada. Por estas razones se encamina el trabajo al uso de las LSTM, ya que aunque existen metodologías más simplificadas, la característica de la memoria es muy atractiva para obtener mayor precisión frente a los cambios atmosféricos. Por otro lado, se ha demostrado que son ágiles en cuanto a cómputo, aunque su comportamiento interno sea más complejo.

3.3 Metodología del trabajo

Para solventar lo expuesto anteriormente se propone un sistema basado en LSTM-RNN como sustitución del controlador clásico PID de un espejo deformable en procesos de Óptica Adaptativa, debido a tres razones principales:

- La predicción se ve afectada por estados anteriores, y lejanos en el tiempo (véase los puntos 2.1 y 2.2). Esto es de gran beneficio para nuestro sistema, ya que la aberración no es más que el movimiento secuencial de la atmósfera. Aunque la atmósfera no es predictiva, sí que puede asumirse cierta continuidad a intervalos muy cortos de tiempo.
- Son robustas ante datos no lineales, y pueden integrar otros tipos de redes dentro de su metodología.

Se puede simplificar la metodología de trabajo a seguir como se refleja en la figura [6]. Dicha metodología se desarrolla de la forma expuesta para poder hacer un estudio completo de los datos con los que se van a trabajar de una manera sencilla y eficaz, de la misma manera que se consigue resolver el problema expuesto.

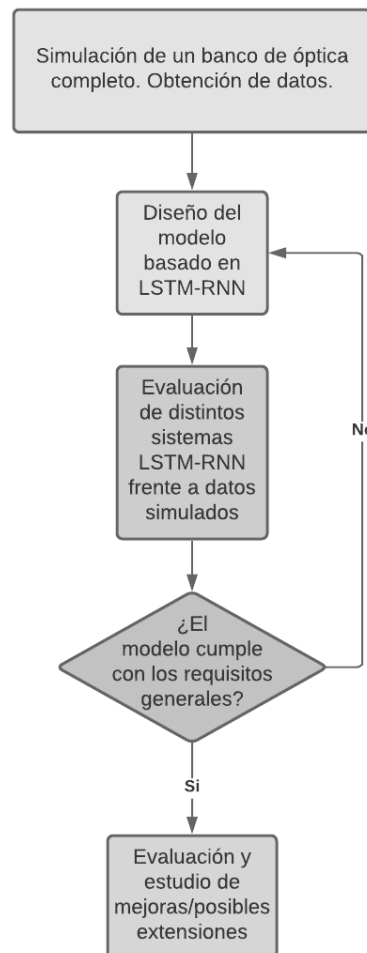


Figura 7. Flujograma de la metodología de trabajo. Fuente: autor

Como primera estancia se deberá trabajar con datos simulados. La razón es meramente para simplificar el trabajo y el desarrollo mientras se entrena y diseña un primer modelo. Gracias al equipo conformado por doctores tanto del Instituto de Astrofísica de Canarias y de la Universidad de Canterbury, se proporciona para este trabajo final de grado diversos sistemas simuladores de estrella y de atmósfera.

Seguidamente de la implementación de la librería ofrecida por el IAC, y su posible adaptación, se encontrará la etapa de estudio de los datos. Esta fase es fundamental para poder realizar un diseño correcto. Posteriormente, se haya una etapa de diseño y evaluación de distintos modelos de LSTM-RNN para resolver la problemática.

Podemos resumir los posibles diseños en los siguientes:

- LSTM-RNN (*Sequence-to-Sequence*), que modela el comportamiento de un PID. [PID-LSTM-RNN]
- LSTM-RNN (*Sequence-to-Sequence*), que no modela directamente un PID, sino que se convierte en un propio controlador

Ambos modelos serán discutidos en el apartado correspondiente. La alimentación del sistema constará de matrices de datos sobre los puntos de desplazamiento calculados con respecto a una posición de referencia de la estrella. Esto implica que trabajaremos con matrices proveniente de imágenes. La salida para ambos casos también será la misma: la conversión de los centroides a modos de zernike. Lo que diferencia ambos modelos no es más que la implementación de una referencia objetivo de nodos de zernike, como sería en el primer caso.

La señal “simulada” de seguimiento para el PID-LSTM-RNN no sería otra que error de los nodos de zernike con respecto el Disco de Airy [24], o, por el contrario, la corrección de los desplazamientos de los centroides. Es decir, se buscaría llevar a un punto cero los centroides calculados de entrada, ya que esto significaría la corrección de la aberración con respecto a la imagen de referencia. En óptica, se conoce el Disco de Airy como el patrón de un haz de luz perfectamente enfocado por una lente ideal. El resultado de la red neuronal, por lo tanto, sería de la aberración contraria para aplanar el frente de onda, acercándose a esa referencia ideal.

En el segundo caso, se desprendería del comportamiento PID a modo de simplificación del sistema. Se estudia el caso de la eliminación del seguimiento del patrón de Airy, para poder predecir la forma de la aberración según tendencias en instantes anteriores.

Como evaluación del correcto funcionamiento de estos, es estrictamente necesario que se cumpla en la mejor medida de lo posible los requisitos generales. Se considerará que el modelo es exitoso si con los datos simulados no demuestra sobreajuste, rápida velocidad de cómputo y robustez ante diversos contextos atmosféricos.

4. Identificación de requisitos

En esta sección se discutirá los datos con los que se va a trabajar para esta metodología, así como el entorno software donde se aplicará el desarrollo y evaluación de la misma. En esta misma sección se hablará de los componentes principales de cada modelo expuesto en el punto 3.3.

4.1 Entorno de desarrollo

Para mayor facilidad de implementación, se trabajará con Jupyter Notebook con la última versión de Python. Además, se utilizarán las siguientes librerías:

- Tensorflow
- Keras
- Librerías de cálculo y representación de datos, como numpy, pandas y matplotlib.

Se requieren extensos conocimientos de Ingeniería Óptica y física para poder simular el comportamiento de las lentes, el frente de onda y las condiciones atmosféricas a nivel algoritmo. Esther Soria, doctorando del proyecto [5] del Instituto de Astrofísica de Canarias [IAC], ofrece un modelo del equipo optomecánico implementado en ALIOLI [5] basado en Python. En dicho modelo, se es capaz de simular el comportamiento de los sensores de onda TP3 y Shack-Hartmann, así como poder realizar simulaciones con distintas perturbaciones atmosféricas. Stephen Weddell, Senior Lecturer y PhD en la Universidad de Canterbury [6] amplía lo introducido por Esther hasta el punto de simular a nivel de fotones. Su librería está implementada, sin embargo, en Matlab.

4.2 Datos

Se deberá trabajar con una extensa cantidad de muestras de imágenes. Se ha señalado que la entrada a este sistema serán matrices de vectores de desplazamiento proporcionados por los sensores de frente de onda respecto a una posición de referencia de la estrella.

Se realizará una base de datos con 4000 muestras, cada muestra constando por una imagen de 265x256 píxeles. Este es un tamaño estándar en la adquisición de imágenes astronómicas. Sin embargo, el tamaño se verá reducido por el empleo de lo que se conoce como *Region of Interest (ROI)*, llegando a evaluarse 178 puntos por imagen.

Los datos de corrección (o targets) son generados por la propia librería proporcionada por el IAC.

Como parte del pre-procesado, es recomendable realizar un estudio estadístico sobre la muestra, así realizar un *feature-scaling* para normalizar los datos de entrada. Esto es muy beneficioso para las redes neuronales, ya que mejora la precisión y la rapidez de cómputo.

También se realizará un *split* del dataset, tal que se disponga de:

- Datos de entrada y salida para training (80% sobre el dataset original)
- Datos Datos de entrada y salida destinados a test (20% sobre el dataset original)

Gracias a esta partición se podrá evaluar de antemano el posible sobreajuste, ya que un modelo que es capaz de tener una precisión casi perfecta (por ejemplo, del 99%) en el training set y posteriormente tenga una precisión pobre respecto al test set, quiere decir que no es capaz de predecir nuevos datos; no es capaz de generalizar. Este fenómeno es el conocido como sobreajuste.

4.3 Modelado de la LSTM-RNN

Esta red será modificada durante el proceso de evaluación descrito en el apartado de metodología. Basada en Keras, se proporciona diversos elementos para evaluar su comportamiento (*F1-Score*, *accuracy*, *recall*, etc). Deberá registrarse el tiempo que emplea por imagen para predecir su forma de onda, así como una descripción de cómo afecta el training de dicho modelo a nivel máquina. Se valorará el mejor balance entre eficiencia y simplificación.

Cabe resumir que deberá tener estas características básicas:

- En la primera capa deberá aceptar un tamaño de datos de entrada predeterminado (numero_de_time_step,tamaño de ROI,1). Las LSTM-RNN se diferencian de las Vanilla RNN en que añaden una dimensión más en los datos de entrada, en este caso, la secuencia temporal.
- La primera capa será de tipo LSTM
- Una capa Dense al final del modelo con una salida para cada nodo de Zernike. Es decir, obtendremos una capa dense con 7 salidas.

4.4 Evaluación de los resultados

Para la evaluación de la metodología se usarán estudios sobre el comportamiento del modelo frente a los datos. Es decir, cuando el modelo recoge en la entrada un frente de onda, ¿Cuánto tiempo pasa desde que recibe los datos hasta que descompone la información en base a nodos de Zernike? ¿Cómo se comporta la computadora, físicamente, durante el proceso? Una valoración sobre los recursos gastados en el momento por la máquina, así como la velocidad y precisión de cálculo, serán partes indispensables de la evaluación del modelo ofrecido en este trabajo.

Para facilitar esta evaluación se usarán recursos implementados ya en Python. A modo de resumen:

- La medida de tiempos se realizará a través de los módulos correspondientes que se hayan en Python, como es el caso de la librería *Time*.
- Cálculo del error cuadrático medio obtenido en la evaluación del modelo
- Gráficas comparativas del coste obtenido por etapa en el entrenamiento, para valorar si los hiperparámetros establecidos son los adecuados

Para facilitar la comprensión sobre los resultados se utilizarán tablas que resuman estos, con las ventajas y desventajas que correspondan a cada caso. Se terminará por escoger el mejor modelo en base a los objetivos expuestos en los apartados correspondientes.

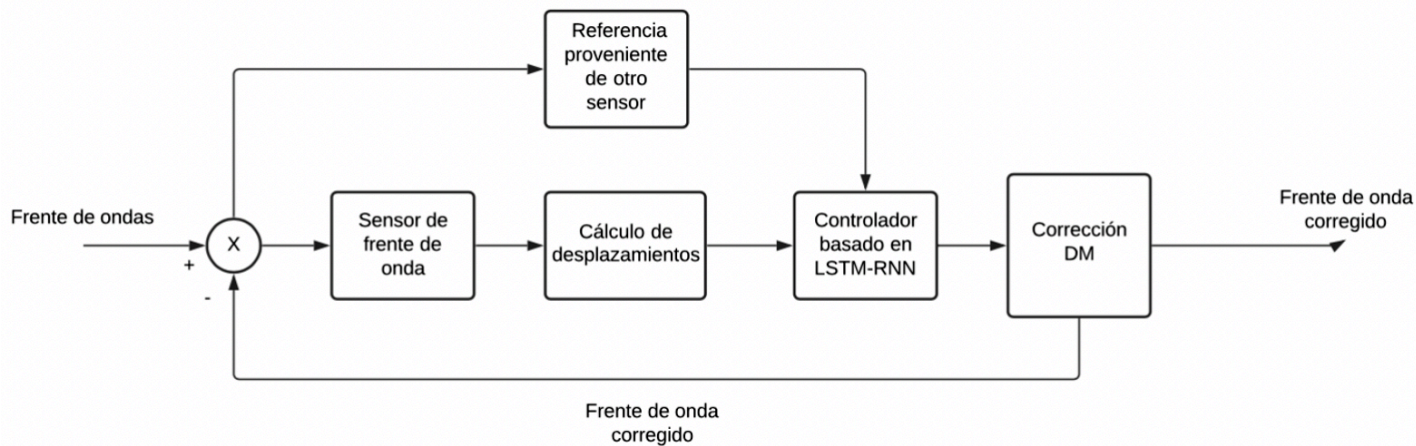


Figura 8. Lazo de control para AO basado en LSTM-RNN. Fuente: Autor

```
In [2]: wavelength_wfs = 700.0E-9
telescope_diameter = 1.5*(0.25/(1.5*13.8))
zero_magnitude_flux = 3.9E10
stellar_magnitude = 0
char= 2
pupil_grid=make_pupil_grid(256, telescope_diameter)
aperture=circular_aperture(telescope_diameter)
TCS_aperture = evaluate_supersampled(aperture, pupil_grid, 1)
```

Figura 9. Definición optomecánica de la fuente de luz y del telescopio. Fuente: Autor / ALIOLI [5]

```
In [5]: star_refx= np.loadtxt('/Users/mcramirez/Desktop/master_ia/TFM/simuladorAI/star_refx.txt')
star_refy= np.loadtxt('/Users/mcramirez/Desktop/master_ia/TFM/simuladorAI/star_refy.txt')
star_regx= np.loadtxt('/Users/mcramirez/Desktop/master_ia/TFM/simuladorAI/star_regx.txt')
star_regy= np.loadtxt('/Users/mcramirez/Desktop/master_ia/TFM/simuladorAI/star_regy.txt')
star_regw= np.loadtxt('/Users/mcramirez/Desktop/master_ia/TFM/simuladorAI/star_regw.txt')
star_regh= np.loadtxt('/Users/mcramirez/Desktop/master_ia/TFM/simuladorAI/star_regh.txt')
```

```
In [6]: star_zdx=np.zeros((np.size(star_refx),num_modes))
star_zdy=np.zeros((np.size(star_refx),num_modes))

for j in range(1,num_modes+1):
    n,m = ansi2dual ansi2dual(j)
    star_zdx[:,j-1]=zernike_dx.zernike_dx(n,m,star_refx,star_refy)
    star_zdy[:,j-1]=zernike_dy.zernike_dy(n,m,star_refx,star_refy)
star_zd=np.zeros((2*np.size(star_refx),num_modes))
star_zd[0:np.size(star_refx),:]=star_zdx
star_zd[np.size(star_refx):,:]=star_zdy
star_zd_pinv=np.linalg.pinv(star_zd)
```

Figura 10. Extracción de datos de la estrella de referencia Fuente: Autor / ALIOLI [5]

```
In [17]: time_series=[]
zernike_modes=[]
list_iter = []
for i in range(num_modes):
    iter_key = "Z"+str(i)
    zernike_modes.append(iter_key)

for i in range(num_iter):
    iter_key = "I"+str(i)
    list_iter.append(iter_key)
```

Figura 11. Extracción de datos de la estrella de referencia Fuente: Autor

```

In [18]: for i in list_iter:
        deformable_mirror.random (0.06* wavelength_wfs)
        wf_dm_a = deformable_mirror.forward(wf)
        wf_wfs= swfs.forward(wf_dm_a)
        detector = subsample_field(wf_wfs.power, subsampling=1, statistic='sum') * exposure_time
        mean = 0.0 # some constant
        std = 0.00005 # some constant (standard deviation)
        detector = detector + np.random.normal(mean, std, detector.shape)
        slopes = SH_est.SH_estimator(detector,star_regx, star_regy, star_regh, star_regw,star_refx,star_refy)
        sim_inputs[i]=slopes
        W=star_zd_pinv.dot(slopes)
        now = datetime.datetime.now()
        string_date=str(now.hour)+"-"+str(now.minute)+"-"+str(now.second)
        time_series.append(string_date)
        for j in range(0,len(W)):
            iter_key="Z"+str(j)
            sim_results[i][iter_key]=W[j]

```

Figura 12. Simulación de los datos. Fuente: Autor / ALIOLI[5]

5. Descripción de la metodología

En esta sección se detallará paso por paso el desarrollo de la metodología. Se hará la descripción junto con figuras explicativas extraídas directamente de código funcional.

La figura [8] representa el lazo de control a nivel software. A nivel opto mecánico seguiría una jerarquía u orden distintos. Se añade la posibilidad de utilizar una señal de referencia, para la opción de expandir el controlador LSTM-RNN a un comportamiento tipo PID.

5.1 Simulación de los datos

Para el diseño de la metodología es necesario generar el conjunto de datos para poder entrenar la red LSTM-RNN. Para ello, se empieza definiendo el contexto óptico.

En el código mostrado en la figura [9], se define las propiedades ópticas de la estrella simulada y las cualidades optomecánicas del telescopio simulado. Estas variables se inicializan con magnitudes estándar, como puede ser por ejemplo la longitud de onda definida. Esta se encuentra en el espectro visible (o en el límite, para mayor exactitud). Como se puede observar, también se ha de definir la pupila, y aquí se inicializa el tamaño de imagen que vamos a obtener en la simulación. Recordando lo mencionado en el apartado 4.2, las imágenes son de tamaño 256x256.

Definiendo el diámetro del telescopio, la apertura, y la apertura del telescopio, estaremos definiendo el camino que atravesará el haz de luz simulado. El resto de las variables son propiedades físicas con relación a la estrella.

Los datos considerados de la estrella de referencia son extraídos directamente de un sensor Shack-Hartmann (Figura [10]). Esta imagen no tiene que ser perfecta, pero sí de la suficiente calidad óptica, ya que los cálculos que se realizarán en base a esta referencia serán los datos de entrada de la LSTM-RNN que se desarrolla en esta metodología.

Para esta estrella se generará una matriz de datos con la posición de cada centroide de referencia. Como última parte de generación de datos simulados, se avanza con la simulación de el haz de la estrella a través del telescopio, y calculándose los desplazamientos de cada instante t simulado con respecto a la estrella de referencia. Los datos de esta estrella de referencia pueden provenir del sensor secundario, a modo de referencia para la opción de modelar el sistema como un controlador PID.

En la figura [11] se muestra cómo se prepara el almacenamiento antes de la simulación de la secuencia del frente de ondas. Se crea una lista de tiempo donde se almacene el instante t de ocurrencia de la perturbación.

Por último, para $n_iteracciones=3000$, y para siete nodos de zernike objetivo, se obtiene:

- Un set de datos con las aberraciones simuladas (`sim_inputs`), en forma de desviación de los centroides con respecto a la imagen de referencia
- Un set de datos con la forma de nodos de zernike de dichas aberraciones (`sim_results`)
- Un set que guarda la secuencia temporal. Para cada instante t de la aberración, se guarda la hora, minuto y segundo de la ocurrencia

Con el algoritmo presentado en la figura [12], ya se tienen los datos necesarios para empezar a trabajar con el modelo.

5.2 Estudio de los datos generados de la simulación

Para mayor facilidad de visualización, se convertirá las listas y diccionarios creados en el paso representado en las figuras [11] y [12] a dataframes, como se puede visualizar en la figura [13].

Para 3000 iteracciones, se ha obtenido 3000 imágenes de 178 puntos. Cada punto representa la variación de los centroides con respecto a la estrella de referencia. Es decir, si se está evaluando la posición en un grid específico, lo que tenemos aquí representado es el gradiente de variación por la perturbación.

```
In [12]: results_dataframe = pd.DataFrame(sim_results)
         inputs_dataframe = pd.DataFrame(sim_inputs)
         series_dataframe = pd.DataFrame({'time': time_series})

In [13]: inputs_dataframe
```

Out[13]:

	I0	I1	I2	I3	I4	I5	I6	I7	I8	I9	...	I3990	I3991	I3992	I39
0	-1.954805	-0.864926	-0.812581	-0.585648	-0.634290	-0.263722	-0.722349	-0.947100	-0.372901	-0.222387	...	-0.569195	-1.175942	0.008611	-1.1313
1	0.282660	-0.233564	-0.885795	-0.654181	0.215604	-0.879876	-0.479568	0.466723	-0.718076	-0.334195	...	-1.155479	-0.350012	-0.223211	-0.5379
2	-0.366342	-0.651429	-0.951800	-0.083862	-1.370890	-0.819204	0.146755	-0.387012	-0.001922	-0.682527	...	-0.253209	-0.654310	0.878741	-0.6453
3	0.023625	-0.064648	-0.197160	-0.867078	-0.457711	0.242958	-0.061789	-0.305492	0.016501	-1.680461	...	-1.125963	0.290133	0.754812	-0.4087
4	-0.018559	-0.871873	-1.055404	-1.345046	-0.424979	-0.510566	-0.744104	-1.109143	-0.455440	-0.848078	...	-1.145438	0.403707	-0.511944	-0.1262
...
173	-0.088444	-1.628757	-0.245820	0.140514	-0.846822	-0.930472	-0.102168	-0.739501	-0.308038	-0.672426	...	-0.240659	0.316895	-0.643437	-0.3577
174	-1.509870	-0.365690	-1.806167	-0.461983	-0.237202	-1.201019	-0.704943	-0.925352	-1.429745	-1.403757	...	-0.297159	-0.745679	-1.132619	-0.1738
175	-1.427911	-0.858679	-0.879967	-0.214421	-0.618866	-1.134719	-1.535912	-0.826339	-0.823967	0.545383	...	-0.310844	-1.357737	-1.558469	-1.3078
176	-0.221038	0.526350	-0.364962	0.144579	0.551146	0.462373	-1.000406	-0.464103	-0.576918	-0.824232	...	1.785124	-0.439036	-0.435646	0.3147
177	-0.109075	-0.800114	-0.439572	0.005243	-0.871320	-0.469290	-0.158414	-0.876407	-1.080815	-0.341679	...	0.135995	-0.432605	-0.664662	-0.7484

178 rows x 4000 columns

Figura 13. Arriba: conversión de los datos obtenidos en la simulación a dataframes. Abajo: dataframe generado como dato de entrada. Fuente: Autor

A su vez se generó un dataset para cada instante t de tiempo en el que se genera cada aberración. Para poder incluir el dataset representado en la figura [14] habrá que realizar una pequeña transformación en los datos, tal y como se señala en la figura [15]. Ahora la secuencia temporal será de $0 < t < 3000$. Podemos así graficar como es el gradiente de aberración para el primer punto del grid de la estrella a lo largo de toda la secuencia temporal, como se refleja en la figura [16].

Con respecto a los valores de salida se obtendrá un dataset de tamaño (número de iteraciones, número de modos de zernike) como se representa en la figura [17]. Los valores que se encuentran en este conjunto de datos complica la precisión del sistema, ya que se trata de valores muy cercanos a cero conforme aumentamos de modo.

In [14]: series_dataframe

Out[14]:

	time
0	18-45-38
1	18-45-38
2	18-45-38
3	18-45-38
4	18-45-39
...	...
3995	18-49-22
3996	18-49-23
3997	18-49-23
3998	18-49-23
3999	18-49-23

4000 rows x 1 columns

Figura 14. Dataset de la secuencia temporal. Fuente: Autor

```
In [31]: time_step = []
        temps = []
        step=0

        for column in inputs_dataframe:
            temps.append(inputs_dataframe[column])
            time_step.append(step)
            step = step + 1
```

Figura 15. Transformación de la secuencia temporal. Fuente: Autor

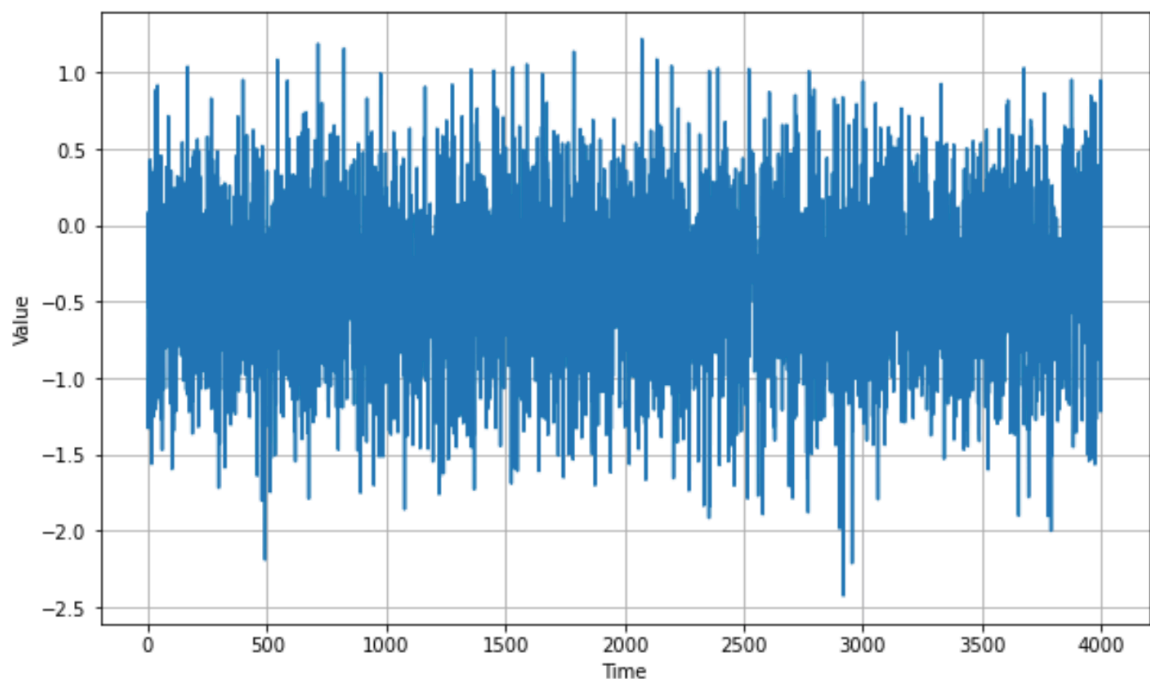


Figura 16. Aberración con respecto a un píxel i,j dentro de una secuencia temporal. Fuente: Autor


```
In [18]: results_dataframe
```

```
Out[18]:
```

	Z0	Z1	Z2	Z3	Z4	Z5	Z6
I0	-0.533847	-0.493512	0.000994	0.000312	0.000181	-1.551898e-06	-3.930457e-07
I1	-0.073878	-0.397619	0.000245	-0.000425	0.001000	-1.403594e-06	7.442946e-07
I2	-0.233618	-0.258434	-0.000226	-0.000269	0.000286	2.234060e-07	5.819669e-07
I3	-0.351788	-0.403345	0.000444	-0.000103	0.000129	-9.261950e-08	6.021987e-09
I4	-0.190162	-0.311956	-0.000032	-0.000519	0.000700	-8.456534e-07	1.071595e-06
...
I3995	-0.239901	-0.285097	-0.000388	-0.000097	0.000208	3.575880e-07	4.620864e-07
I3996	-0.363583	-0.104100	-0.000365	0.000261	-0.000621	6.109373e-07	-3.714135e-07
I3997	-0.283131	-0.484438	0.000390	-0.000055	0.000683	-1.031399e-06	2.765321e-07
I3998	-0.086875	-0.545973	0.000594	-0.000478	0.001494	-1.676468e-06	9.483792e-07
I3999	-0.307501	-0.171271	-0.000448	-0.000009	-0.000423	1.071578e-06	3.260466e-08

4000 rows x 7 columns

Figura 17. Dataset de la representación del frente de onda para cada imagen i respecto a los siete primeros nodos de Zernike. Fuente: Autor

```
In [19]: inputs_dataframe.describe()
```

```
Out[19]:
```

	I0	I1	I2	I3	I4	I5	I6	I7	I8	I9	...	I3990	I3991
count	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	...	178.000000	178.000000
mean	-0.535884	-0.539661	-0.573345	-0.585609	-0.601295	-0.567420	-0.566510	-0.627796	-0.535121	-0.556136	...	-0.518126	-0.522237
std	0.569750	0.562479	0.549828	0.581719	0.572125	0.584310	0.558786	0.559670	0.590854	0.608714	...	0.591485	0.577340
min	-2.046295	-1.970958	-2.090861	-1.809786	-2.159935	-2.363197	-1.805083	-2.111394	-2.247856	-1.919574	...	-1.682114	-2.626952
25%	-0.892538	-0.917450	-0.963175	-0.958986	-0.976582	-0.980117	-0.983329	-0.980978	-0.868404	-0.938248	...	-0.938119	-0.870638
50%	-0.495740	-0.587527	-0.549791	-0.614188	-0.600997	-0.572176	-0.566409	-0.638361	-0.554314	-0.607858	...	-0.593152	-0.519841
75%	-0.159330	-0.209118	-0.177185	-0.232486	-0.231225	-0.231077	-0.151622	-0.246406	-0.170242	-0.154834	...	-0.089419	-0.145728
max	1.336324	1.191931	0.841351	0.814295	0.789757	0.864481	0.769699	0.843383	1.271633	1.326334	...	1.785124	1.005590

8 rows x 4000 columns

Figura 18. Estudio estadístico del dataset de entrada. Fuente: Autor

Podemos averiguar un poco más sobre los datos haciendo uso de la propiedad `dataframe.describe()`, método diseñado para estudiar las propiedades estadísticas de un dataset.

Como se puede visualizar en la figura [18], debido a que no se ha sometido el sistema óptico a una aberración muy compleja, el sistema tiende a una distribución normalizada. Para poder generalizar mejor el uso del método de LSTM-RNN que se presenta en este trabajo, y para mejorar el funcionamiento del mismo, se normalizará el dataset de entrada, indiferentemente de si se realiza partición para la parte de validación o para la parte de entrenamiento.

In [89]:

```
model_lstm = Sequential()
model_lstm.add(LSTM(178, input_shape = (1, 178), return_sequences=True))
model_lstm.add(Dense(num_modes))
model_lstm.summary()
```

Model: "sequential_3"

Layer (type)	Output Shape	Param #
lstm_3 (LSTM)	(None, 1, 178)	254184
dense_2 (Dense)	(None, 1, 23)	4117

Total params: 258,301
 Trainable params: 258,301
 Non-trainable params: 0

In [91]:

```
plot_model(model_lstm, to_file='model_plot.png', show_shapes=True, show_layer_names=True)
```

Out[91]:

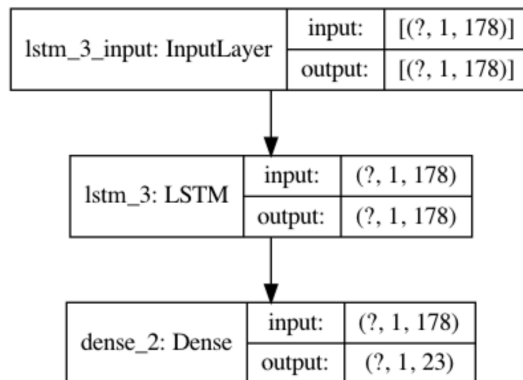


Figura 19.-Modelo LSTM-RNN aplicado a AO. Fuente: Autor

5.3 Diseño y entrenamiento

Gracias al soporte ofrecido por Keras en Python, el diseño de la RNN es bastante simplificado. Un modelo simplificado de una LSTM-RNN aplicado a Óptica Adaptativa se puede encontrar en la figura 19.

Como se puede observar, se utilizarán 178 filtros en la entrada (uno por cada input), y una capa final con 23 filtros (uno por cada salida deseada). Para demostrar la capacidad de esta metodología, se evaluará el comportamiento del modelo con la totalidad de nodos de Zernike.

Este modelo va a ser utilizado tanto como para la evaluación con y sin PID. El modelo en PID consistirá de un paso más en el tratamiento de los datos anterior a la alimentación de la RNN. Esto servirá, por lo tanto, para evaluar que modelización es más precisa y computacionalmente menos costosa.

```
In [600]: model = MLPRegressor(
            hidden_layer_sizes=(100,),
            learning_rate='constant',
            activation='tanh',
            alpha=0.05,
            solver='adam',
            max_iter=25
        )
```

Figura 20. Modelo ANN simplificado para AO. Fuente: Autor

```
In [578]: mlp = MLPRegressor(hidden_layer_sizes=(10,10,10), max_iter=1000)
            parameter_space = {
                'hidden_layer_sizes': [(50,50,50), (50,100,50), (100,)], 'activation': ['tanh', 'relu'],
                'solver': ['sgd', 'adam'],
                'alpha': [0.0001, 0.05],
                'learning_rate': ['constant', 'adaptive'], }
            clf = GridSearchCV(mlp, parameter_space, n_jobs=-1, cv=3)
```

```
In [580]: clf.fit(Xtrain,ytrain)
```

```
Out [580]: GridSearchCV(cv=3,
                        estimator=MLPRegressor(hidden_layer_sizes=(10, 10, 10),
                                                max_iter=1000),
                        n_jobs=-1,
                        param_grid={'activation': ['tanh', 'relu'],
                                    'alpha': [0.0001, 0.05],
                                    'hidden_layer_sizes': [(50, 50, 50), (50, 100, 50),
                                                            (100,)],
                                    'learning_rate': ['constant', 'adaptive'],
                                    'solver': ['sgd', 'adam']})
```

```
In [454]: print('Best parameters:\n', clf.best_params_)
```

```
Best parameters:
{'activation': 'tanh', 'alpha': 0.05, 'hidden_layer_sizes': (100,), 'learning_rate': 'constant', 'solver': 'adam'}
```

```
In [600]: model = MLPRegressor(
            hidden_layer_sizes=(100,),
            learning_rate='constant',
            activation='tanh',
            alpha=0.05,
            solver='adam',
            max_iter=25
        )
```

Figura 21. GridSearch en el modelo simplificado ANN. Fuente: Autor

Cabe mencionar que se ha diseñado una metodología LSTM bidireccional. Según [28], las redes de este tipo pueden mejorar notoriamente los procesos de identificación de patrones en secuencias de datos. Es por ello que en la figura 19 (imagen inferior) se aprecia una segunda capa LSTM, aunque esta actuaría como una oculta.

Como comparación, se realizará un tercer modelo basado en redes neuronales simples, aplicadas como un regresor lineal. En la figura 20 se puede visualizar el diseño de este modelo.

```
In [646]: X_train, X_test, y_train, y_test = train_test_split(inputs_dataframe.transpose(),  
                                                           results_dataframe.transpose(), test_size=0.3, random_state=109)
```

Figura 22. Split del conjunto de datos para entrenamiento y validación de los modelos propuestos.

Fuente: Autor

Este modelo ha sido generado a partir de la realización de GridSearch con relación a un conjunto de posibles hiperparámetros. GridSearchCV es una función de Keras que se encarga de evaluar el modelo dentro de un conjunto de hiperparámetros, y devuelve aquellos con los que se ha obtenido mejores resultados. En la figura 21 se puede observar este procedimiento.

5.3.1 Preparación previa de los datos

Para trabajar con los datos se necesitará agruparlos en partes diferenciadas: en este caso, en *training set* y *test set*. Un porcentaje de la base de datos real de entrada se guardará como datos de entrenamiento, mientras que los datos restantes se agruparán como datos de prueba. Los datos de entrenamiento (como su propio nombre indica), serán utilizados durante la fase de entrenamiento de la red neuronal. Esta práctica se realiza en Aprendizaje Automático para ayudar a evitar el temido sobreajuste a los datos, y así poder valorar la capacidad predictiva del modelo creado.

En Python, este “split” en el conjunto de datos de entrada se realiza como se muestra en la figura 22.

Lo siguiente que se deberá hacer es una estandarización de los datos. El objetivo con esta práctica será normalizar los datos con los que se trabajarán, ya que así se consigue una mejor modelización de los mismos. Los datos de entrada siguen una distribución uniforme, esta estandarización no sería obligatoria en estos, pero donde será crucial será sobre los datos de salida.

Los modos de Zernike no siguen una misma distribución para cada nivel. Podemos encontrar una magnitud muy distinta entre los modos 1 y 20 (se debe recordar que cuanto más alto es el modo de zernike más tiende a cero su valor), por lo que una red neuronal puede dar fallos de cálculo al intentar modelar la salida del sistema. Para evitar dicho problema, se recurre a su normalización, pudiéndose deshacer la misma con posterioridad para obtener la predicción real. En la figura 23 se puede apreciar este paso.

```
In [647]: from sklearn.neural_network import MLPRegressor
scaler = StandardScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
scaler_o = StandardScaler()
scaler_o.fit(y_train)
y_train = scaler_o.transform(y_train)
y_test = scaler_o.transform(y_test)
```

Figura 23. Normalización del conjunto de datos. Fuente: Autor

```
In [648]: X_test=X_test.reshape(X_test.shape[0],1,178)
X_train=X_train.reshape(X_train.shape[0],1,178)
```

```
In [650]: data_y=y_train.reshape(y_train.shape[0],1,num_modes)
test_y =y_test.reshape(y_test.shape[0],1,num_modes)
```

Figura 24. Redimensionalización del conjunto de datos. Fuente: Autor

Por último, para poder trabajar con redes tipo LSTM, hace falta añadir una tercera dimensionalidad en los datos. Esto se consigue reestructurando los conjuntos de la forma presentada en la figura 24.

Sin embargo, este paso no es necesario para trabajar con el modelo basado en ANN. En el caso del uso de la semejanza con un PID, hay que añadir otro tratamiento previo, modelando la entrada del sistema de la siguiente forma:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de}{dt} \quad (7)$$

Donde:

- U(t) es la variable de control
- K_p es la ganancia proporcional
- K_d es la ganancia derivativa
- K_i es la ganancia integradora
- E(t) es la señal de error

```
In [73]: # PID Parameters
Kc = 1.0
tauI = 15.0 # sec
tauD = 0.0 # sec

KP = Kc
KI = Kc / tauI
KD = Kc * tauD

err_signal = err - X_train
zeros = np.zeros((X_train.shape[0], 1, 1))
x_int = integrate.cumtrapz(err_signal)
it = np.append(x_int, zeros, axis=2)
PID = KP*err_signal + KI*it
```

Figura 25. Modelado del conjunto de datos como un control PID. Fuente: Autor

Se entiende como señal de error como la diferencia de la señal actual entre la deseada. Así pues, el controlador PID intentará ajustar el sistema para que siga la señal de referencia. En nuestro caso, la señal deseada será los centroides desplazados a la posición de referencia; o, dicho de otro modo, deshacer el cambio de desplazamientos.

Sin embargo, no se utilizará la parte derivativa del controlador PID, ya que podría amplificar el error y provocar inestabilidad en el control. Se simplificará, por lo tanto, a una aproximación PI de control.

Tras una serie de pruebas, se ha dejado los valores del controlador de la siguiente forma:

- $K_p = 1$
- $K_i = 0.067$
- $K_d = 0$

Lográndose una métrica de error muy cercana a la que se consigue con una LSTM más sencilla, como se presentará en la evaluación final, y un debate sobre su uso o no en cuanto a las mejoras que presente.

Para lograr esta aproximación, Python permite directamente trabajar con el sistema matricial. La parte integradora se ha obtenido al usar una librería específica del entorno de Scipy, un módulo llamado *Integrate*. Dicho módulo ha permitido integrar la matriz de datos de entrenamiento dentro de una secuencia temporal.

Posteriormente, se le aplicará el proceso descrito en la figura 25 para poder utilizar este conjunto “PID” como alimentación de la red neuronal.

```
In [70]: model_lstm.compile(loss='mean_squared_error', optimizer="adam",
                        metrics=['mean_absolute_error', 'mean_squared_error', 'accuracy'])
```

Figura 26. Compilación del modelo LSTM. Fuente: Autor

```
In [71]: start = time.time()
        history = model_lstm.fit(X_train, data_y, epochs=100, validation_split = 0.2, shuffle=False,
                                end = time.time())

53/53 [=====] - 0s 3ms/step - loss: 3.2764e-04 - mean_absolute_er
ror: 0.0144 - mean_squared_error: 3.2764e-04 - accuracy: 0.9548 - val_loss: 0.0252 - val_m
ean_absolute_error: 0.1163 - val_mean_squared_error: 0.0252 - val_accuracy: 0.8143
Epoch 37/100
53/53 [=====] - 0s 3ms/step - loss: 3.5393e-04 - mean_absolute_er
ror: 0.0150 - mean_squared_error: 3.5393e-04 - accuracy: 0.9577 - val_loss: 0.0255 - val_m
ean_absolute_error: 0.1170 - val_mean_squared_error: 0.0255 - val_accuracy: 0.8190
Epoch 38/100
53/53 [=====] - 0s 3ms/step - loss: 4.1452e-04 - mean_absolute_er
ror: 0.0161 - mean_squared_error: 4.1452e-04 - accuracy: 0.9595 - val_loss: 0.0259 - val_m
ean_absolute_error: 0.1182 - val_mean_squared_error: 0.0259 - val_accuracy: 0.8167
Epoch 39/100
53/53 [=====] - 0s 3ms/step - loss: 5.3145e-04 - mean_absolute_er
ror: 0.0180 - mean_squared_error: 5.3145e-04 - accuracy: 0.9548 - val_loss: 0.0265 - val_m
ean_absolute_error: 0.1199 - val_mean_squared_error: 0.0265 - val_accuracy: 0.8310
Epoch 40/100
53/53 [=====] - 0s 3ms/step - loss: 7.4177e-04 - mean_absolute_er
ror: 0.0211 - mean_squared_error: 7.4177e-04 - accuracy: 0.9548 - val_loss: 0.0275 - val_m
ean_absolute_error: 0.1223 - val_mean_squared_error: 0.0275 - val_accuracy: 0.8071
Epoch 0040: early stopping
```

Figura 27. Training del modelo. Fuente: Autor

5.3.2 Entrenamiento

Es en esta fase donde la red calcula los pesos apropiados para poder predecir los modos de Zernike. Estos pesos se van ajustando durante el entrenamiento para poder alcanzar los objetivos propuestos en el modelo.

Antes de realizar este proceso, se debe compilar el modelo (véase figura 26). Es en este paso donde se define la función de coste que se va a utilizar para evaluar el modelo, así como los optimizadores a utilizar y las métricas de error necesarias.

En Keras se presentan las siguientes opciones de funciones de coste para regresión:

- Mean Squared Error
- Mean Absolute Error
- Huber function

Entre otras. Para este modelo se ha escogido la función de Error Cuadrático Medio, ya que se utiliza para verificar cómo de cerca se encuentra un valor estimado al valor real. Cuanto más reducido es este valor, más cercano es al valor real [29].

Para la parte de optimización, se dispone de los siguientes modelos:

- Adam
- SGD
- RMSprop

Entre otros. De estos modelos, el que mejor funcionó durante el entrenamiento fue Adam. Este optimizador, según [30], es computacionalmente eficiente, requiere de poca memoria, y es perfecto para problemas de gran densidad de datos y parámetros. Parece justificada su eficiencia, por lo tanto, sobre el ajuste al patrón de datos proveniente del sensor de frente de onda. Dejando sus valores por defecto, se adquirieron muy buenos resultados durante el entrenamiento, como se expondrá a continuación.

Para el entrenamiento, se va a utilizar la tecnología conocida como “*Early stopping*”, en Keras. Esta tecnología permite evaluar la red neuronal, y frena el entrenamiento cuando el modelo llega a una convergencia en la función de coste. Es decir, este método sirve para ajustar el número de epochs mínimo con máxima efectividad. Esto se logra como se muestra en la figura 27.

Sobre el entrenamiento, cabe mencionar la enorme rapidez con la que el modelo LSTM ha logrado entrenar. Para el modelo LSTM+PID, el rendimiento es muy similar. Los resultados se evaluarán en el capítulo seis de este documento.

También se ha de mencionar que se agrega el parámetro de “*validation-split*” para que la red neuronal sea capaz de autoevaluarse durante esta fase de entrenamiento. El 20% de los datos de entrada se utilizarán con este fin.

Por último, cabe mencionar que el modelo que se utiliza de comparación (es decir, la red ANN), se entrena de una manera más simplificada, tal que, tras el *GridSearch*, basta con ejecutar la orden de *fit model*.

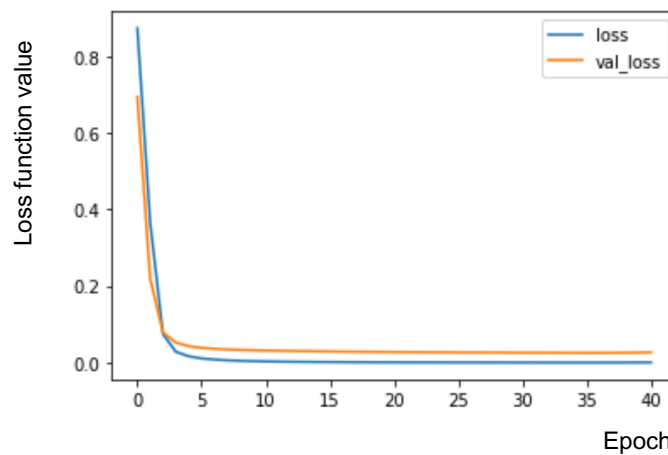


Figura 28. Curva de pérdidas por epoch del modelo LSTM. Fuente: Autor

6. Evaluación

Tras el entrenamiento de los modelos, se examinarán en este capítulo sus resultados, discutiendo sobre la precisión y rapidez de cada uno de ellos, validando la metodología de usar LSTM para óptica Adaptativa.

Se debe mencionar que para la evaluación final se ha dispuesto del uso completo de todos los nodos de Zernike. Es decir, se produce una secuencia de 23 parámetros en la salida de los modelos propuestos como resolución.

6.1 Modelo LSTM

Una vez completado el entrenamiento, se puede realizar un informe de valoración del comportamiento del modelo creado. Este puede empezar por graficar la reducción de la función de coste por epoch, como se representa en la figura 28. En dicha figura se representa las curvas de pérdidas tanto para el *training set* como para el *validation set*.

Gracias a esta gráfica, se puede deducir si se está produciendo sobreajuste en los datos, o en su defecto, *underfitting*. El caso de *underfitting* se produce cuando las pérdidas no se logran reducir visiblemente a lo largo de todas las iteracciones, sino que se mantiene en un valor elevado casi constante.

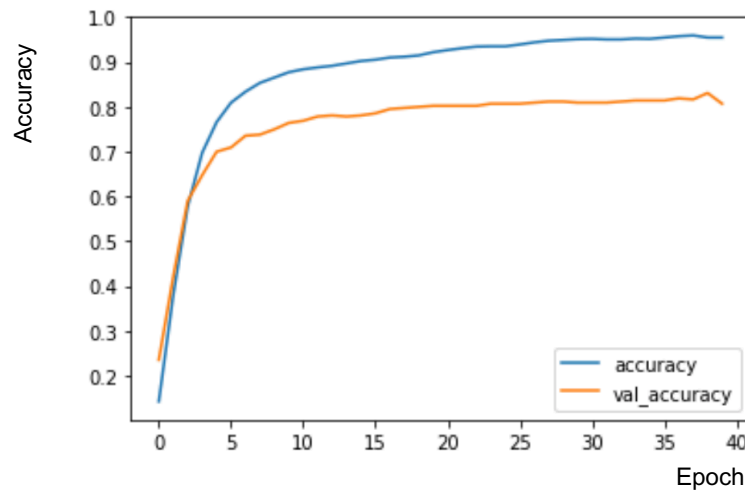


Figura 29. Precisión sobre la validación. Fuente: Autor

Por el contrario, un caso de sobreajuste daría la gráfica de training con los valores que se pueden visualizar en la figura 28, con la excepción de que el coste de validación se incrementaría por iteracción, en lugar de reducirse. Esto querría decir que la red neuronal tendría problemas para ajustarse a nuevos valores, pero como se puede comprobar este no es el caso. Para el modelo propuesto de LSTM, se puede defender de que los resultados del entrenamiento han sido excelentes.

Por otro lado, se puede graficar los resultados de la precisión durante la evaluación del modelo, que se pueden ver reflejados en la gráfica representada en la figura 29.

En la figura 30 se muestra la medida temporal que ha consumido la red LSTM para predecir 900 muestras, de 178 features, produciendo una salida de 23 nodos de Zernike. En la figura 31 se muestra el valor del error cuadrático medio obtenido.

```
In [84]: start=time.time()
test_predict=model_lstm.predict(X_test)
end = time.time()
print("Elapsed time in predict: {:.2f} seconds".format(end-start))

Elapsed time in predict: 0.06 seconds
```

Figura 30. Resultado temporal modelo LSTM. Fuente: Autor

```
In [90]: test_predict=scaler_o.inverse_transform(test_predict)
y_test = scaler_o.inverse_transform(y_test)
test_predict=test_predict.reshape(len(test_predict),num_modes)

In [97]: testScore = math.sqrt(mean_squared_error(y_test[:,0], test_predict[:,0]))
print('Test Score RMSE:',testScore)

Test Score RMSE: 6.0693143865645466e-06
```

Figura 31. Valoración del error en el modelo LSTM. Fuente: Autor

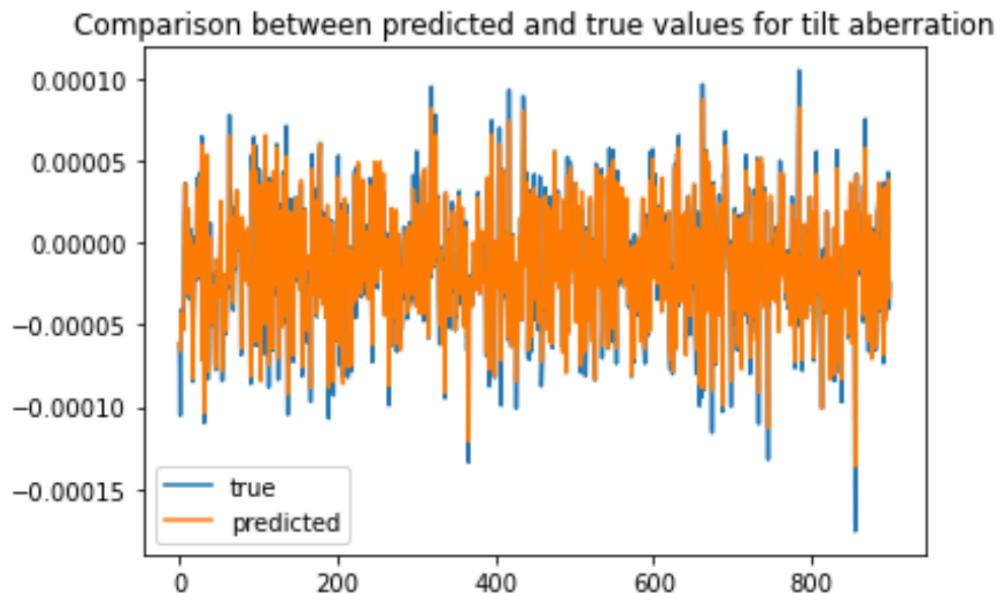


Figura 32. Comparación entre los valores calculados y verdaderos para la aberración tipo tilt. Fuente: Autor

Como se puede apreciar en la figura 31, se realiza una “desconversión” de la normalización realizada como paso previo antes del training. Gracias a esto, obtenemos las medidas reales de los modos de zernike, como se muestra en la figura 32. En la figura 31, también se puede apreciar el bajísimo valor del error RMSE. Gráficamente, esto se puede comprender con mayor facilidad superponiendo la gráfica de secuencialidad para la aberración de tipo tilt “real” contra la calculada por la red neuronal de LSTM. Estas gráficas para cada nodo de Zernike se podrán contemplar en los anexos, en su apartado correspondiente. A continuación, en la tabla 3, se resumirá los aspectos más relevantes de este modelo.

Tabla 3.-Valoración modelo LSTM

Parámetro	Valor
Precisión pico - Validation set	0.83095
Precisión pico - Training set	0.95952
Coste final – Validation set	0.025
Coste final – Training set	0.0003218
Velocidad de procesado por matriz (ms/matriz)	0.06
Tiempo de entrenamiento (segundos)	10.46
Valor RMSE	6.069 e-06

Fuente: Autor

Estos valores cumplen con los requerimientos necesarios para trabajar con precisión y bajo altas prestaciones en Óptica Adaptativa.

6.2 Comparación con otros modelos

En este apartado se discutirá los resultados con los modelos LSTM-PID y ANN. Se expondrá los resultados para cada modelo utilizando el mismo procedimiento descrito anteriormente. En este modelo se puede apreciar una tendencia a la inestabilidad a medida que se incrementan las iteraciones, por lo que el problema puede ser el ajuste de las ganancias. Sin embargo, los resultados son competentes y merecen ser expuestos.

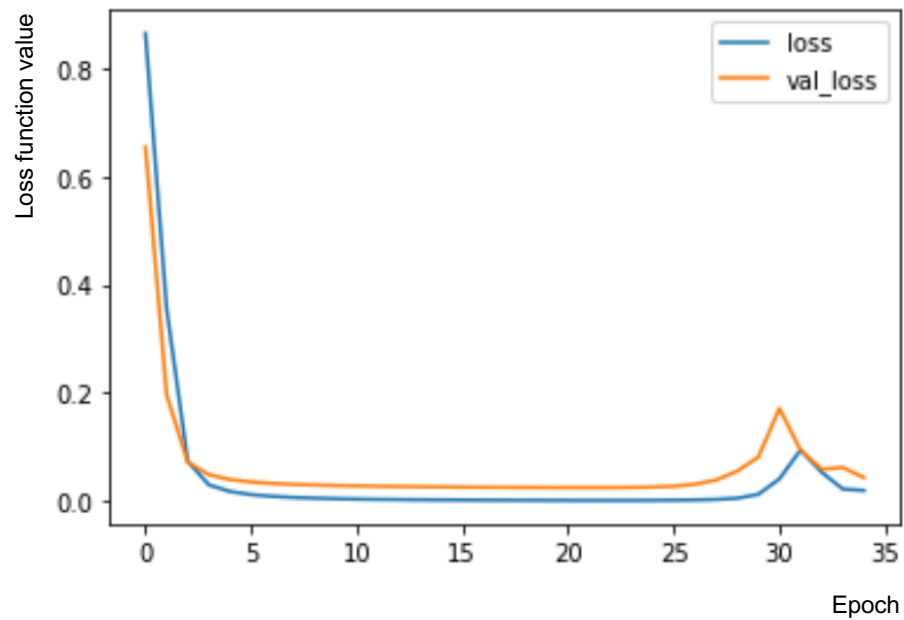
Modelo LSTM-PID

Figura 33. Gráfica de función de coste para LSTM+PID. Fuente: Autor

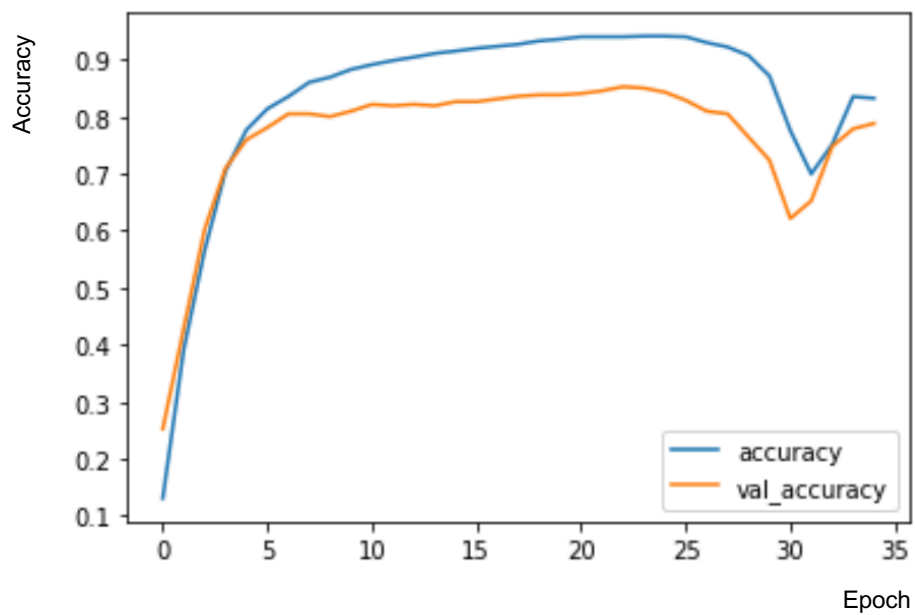


Figura 34. Gráfica de precisión para LSTM PID. Fuente: Autor

Tabla 4. Valoración modelo LSTM mas PID

Parámetro	Valor
Precisión pico - Validation set	0.833
Precisión pico - Training set	0.93
Coste final – Validation set	0.028
Coste final – Training set	0.00155
Velocidad de procesamiento por matriz (ms/matriz)	0.06
Tiempo de entrenamiento (segundos)	8.115
Valor RMSE	4.36e-04

Fuente: Autor

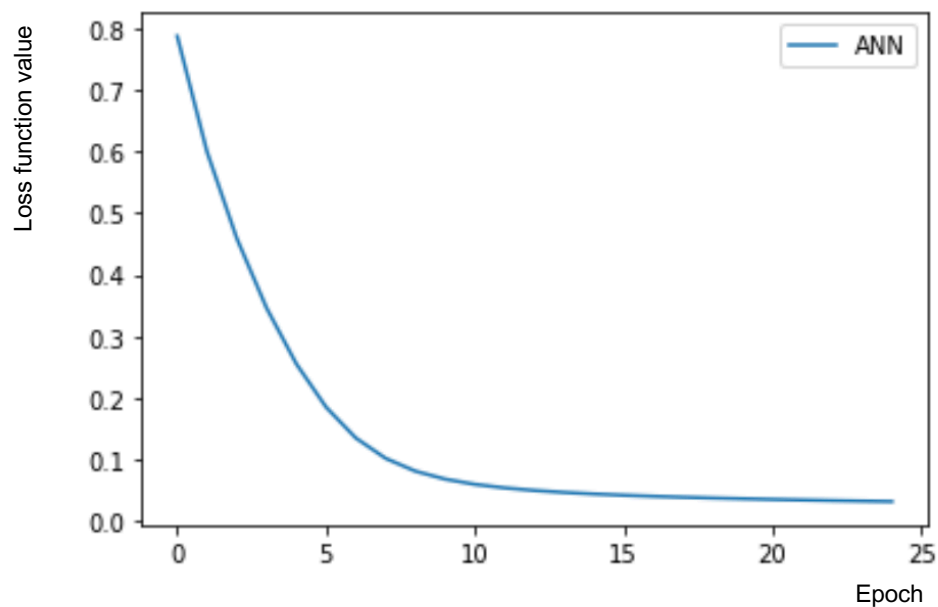
Modelo ANN

Figura 35. Curva de pérdidas del modelo ANN

Tabla 5.-Resultados del modelo ANN

Parámetro	Valor
Score	0.95627
Coste final	0.0311
Velocidad de procesado por matriz (ms/matriz)	0.05
Tiempo de entrenamiento (segundos)	0.3026
Valor RMSE	4.38e-04

Fuente: Autor

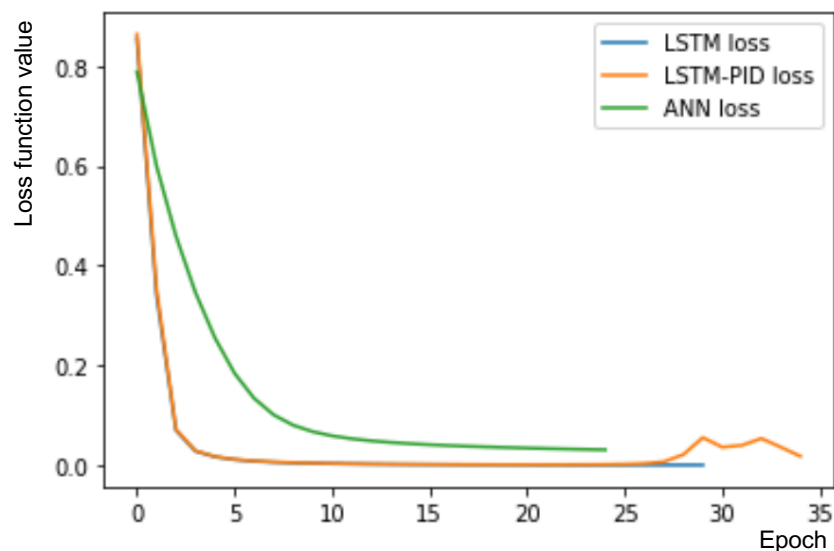


Figura 36. Comparativa de costes entre modelos. Fuente: Autor

En la figura 35 podemos observar que este modelo de redes neuronales es más lento en converger que los otros dos modelos, por lo que necesitaría más iteraciones para llegar a los mismos resultados que los otros modelos, pero es igualmente competente. La comparación final deja una difícil decisión a simple vista, pero en la figura 36 se refleja la superposición de las gráficas de funciones de costes para los tres modelos.

Reduciendo a metodologías más precisas, se descarta el uso del modelo ANN. En la convergencia se demuestra más lenta, aunque los valores de precisión compitan con el modelo LSTM. También carece de la seguridad que ofrece la memoria de la LSTM para la

adaptabilidad de imprevistos en las aberraciones, por lo que aunque es un apto competidor, se descarta su uso para esta aplicación.

Por otro lado, entre los modelos LSTM y LSTM con PID, LSTM se demuestra que presenta mejores prestaciones que un modelo basado en PID, por lo que esto corrobora a que se puede prescindir de un modelo de control PID para esta aplicación. Esto simplifica el tratamiento de los datos, y el tiempo general desde que se adquieren las imágenes (los datos en forma matricial), hasta que se descomponen en nodos de Zernike.

7. Conclusiones y trabajo futuro

En este apartado se resumirá la metodología presentada, discutiendo su utilidad, finalidad y resultados. Por último, se hablará de líneas de trabajo futuro.

7.1 Conclusiones

En este trabajo final de máster se ha discutido, diseñado y evaluado el reemplazo de un controlador típico industrial PID empleado durante el proceso de Óptica Adaptativa por metodologías propias del Aprendizaje Automático. Debido al tipo de datos que se requieren para lograr dicho fin, se redujo las metodologías a aquellas propias de Sistemas Cognitivos.

Se ha realizado una simulación en Python del banco opto-mecánico para facilidad de estudio. La metodología ha sido evaluada sobre la plataforma Jupyter Notebook, y se puede encontrar el resto de información en los respectivos anexos.

Se han evaluado modelos que pudieran ofrecer una buena calidad de predicción de manera rápida y eficiente, como son las LSTM y las feed-forward network. Tras una extensa evaluación, se concluye que las más precisas y rápidas para los requerimientos exigidos en observatorio son cumplidos por el modelo LSTM simple, aunque se demuestra su efectividad también con otros modelos, aunque no sean los óptimos en cuanto a cálculo, ya que pueden precisar de más tratamiento anterior a la evaluación o no poseen las ventajosas características que presentan las LSTM.

A modo resumen, se presenta un modelo LSTM simplificado capaz de:

- Evaluar los 23 nodos de Zernike a partir de un área de interés predefinido en el sensor de frente de onda, costando de 178 puntos como “features” en el conjunto de datos. Cada punto representa un vector de desplazamiento respecto a la posición de referencia de la estrella.
- Es capaz de ajustarse a nuevas aberraciones con éxito y precisión, y gracias a su memoria somos capaces de poder garantizar su adaptabilidad a nuevos casos, lo cual realza su seguridad.
- Increíblemente rápida. Sus resultados en este ámbito han superado muchas expectativas de funcionamiento.

Con lo cual, se concluye que se puede reemplazar el sistema de control original por un sistema mucho más efectivo y rápido gracias a la Inteligencia Artificial. Cabe destacar que, el sistema anterior dependía de muchos módulos distintos para llegar a un resultado parecido, por lo que se logra con esta metodología no solo mejorar la precisión y rapidez del sistema, sino también los costes.

7.2 Líneas de trabajo futuro

Aunque se ha logrado eliminar un principal componente del sistema opto-mecánico y así agilizar el proceso, aun podría verse increíblemente mejorada esta técnica si se ampliase a la modelización de un frente de onda a través de la imagen que proviene directamente de la cámara de ciencia, en lugar de la correspondiente del sensor.

Esto eliminaría el uso de un sensor de frente de onda utilizado en el control de lazo cerrado del sistema. Constaría, con este paso, de un sistema en lazo abierto con una red convolucional de alto rendimiento que podría solventar todos los problemas descritos en este trabajo final de máster. Pero dicho diseño se escapa de los límites temporales de este trabajo, pudiendo ser un trabajo digno de un investigador doctoral.

Muchas de las referencias aportadas en este trabajo apoyan esa teoría, aunque se aplicaban en su mayoría a pequeños módulos dentro del procesado. Aun no se ha hecho efectivo un sistema completamente libre de sensores.

8. Bibliografía

- [1] Friedlander, M. W. (n.d.). Astronomy - The techniques of astronomy. Encyclopedia Britannica. Recuperado el 26 de abril del 2021, en <https://www.britannica.com/science/astronomy/The-techniques-of-astronomy#ref910526>
- [2] Mackay, C. (n.d.). High-resolution imaging with large ground-based telescopes. Spie.Org. Recuperado el 4 de mayo del 2021, en <https://spie.org/news/5408-high-resolution-imaging-with-large-ground-based-telescopes?SSO=1>
- [3] information@eso.org. (n.d.). Adaptive Optics. ESO. Recuperado el 4 de mayo del 2021, en https://www.eso.org/public/teles-instr/technology/adaptive_optics/
- [4] Leigui de Oliveira, M., & Gabriel, J. P. S. (2018). The characterization of the Zernike modes at the focal plane for Extremely Large Telescope projects. Adaptive Optics Systems VI, 2–3. <https://doi.org/10.1117/12.2315537>
- [5] AOLI-Presentation. (n.d.). Copyright 2021 AOLI. All Rights Reserved. (Http://Research.iac.es/Proyecto/AOLI). <http://research.iac.es/proyecto/AOLI/>
- [6] The University of Canterbury. UC Research Profile - University of Canterbury - New Zealand. <https://researchprofile.canterbury.ac.nz/Group.aspx?groupid=45>
- [7] Wizinowich, P. (2015). Adaptive optics in astronomy. Contemporary Physics, 56(4), 432–450. <https://doi.org/10.1080/00107514.2015.1041765>
- [8] Edmund Optics. (2021). Introduction to Adaptive Optics and Deformable Mirrors. Edmund Optics Inc. <https://www.edmundoptics.com/knowledge-center/application-notes/optics/introduction-to-adaptive-optics-and-deformable-mirrors/>
- [9] Colodro-Conde, C., Velasco, S., Fernández-Valdivia, J. J., López, R., Oscoz, A., Rebolo, R., Femenía, B., King, D. L., Labadie, L., Mackay, C., Muthusubramanian, B., Pérez Garrido, A., Puga, M., Rodríguez-Coira, G., Rodríguez-Ramos, L. F., Rodríguez-Ramos, J. M., Toledo-Moreo, R., & Villó-Pérez, I. (2017). Laboratory and telescope demonstration of the TP3-WFS for the adaptive optics segment of AOLI. Monthly Notices of the Royal Astronomical Society, 467(3), 2855–2868. <https://doi.org/10.1093/mnras/stx262>
- [10] van Dam, M. A., & Lane, R. G. (2002). Extended analysis of curvature sensing. Journal of the Optical Society of America A, 19(7), 1390. <https://doi.org/10.1364/josaa.19.001390>
- [11] Education, I. C. (7 de Abril del 2021). Recurrent Neural Networks. IBM. <https://www.ibm.com/cloud/learn/recurrent-neural-networks>
- [12] Search the world's largest collection of optics and photonics applied research. (n.d.). SPIE Digital Library. <https://www.spiedigitallibrary.org>
- [13] Pedro Escárate, Gustavo Rahmer, and Julian Christou "Neural network wavefront sensor for the Large Binocular Telescope Observatory", Proc. SPIE 11448, Adaptive Optics Systems VII, 114484F (13 de diciembre 2020); <https://doi.org/10.1117/12.2562406>

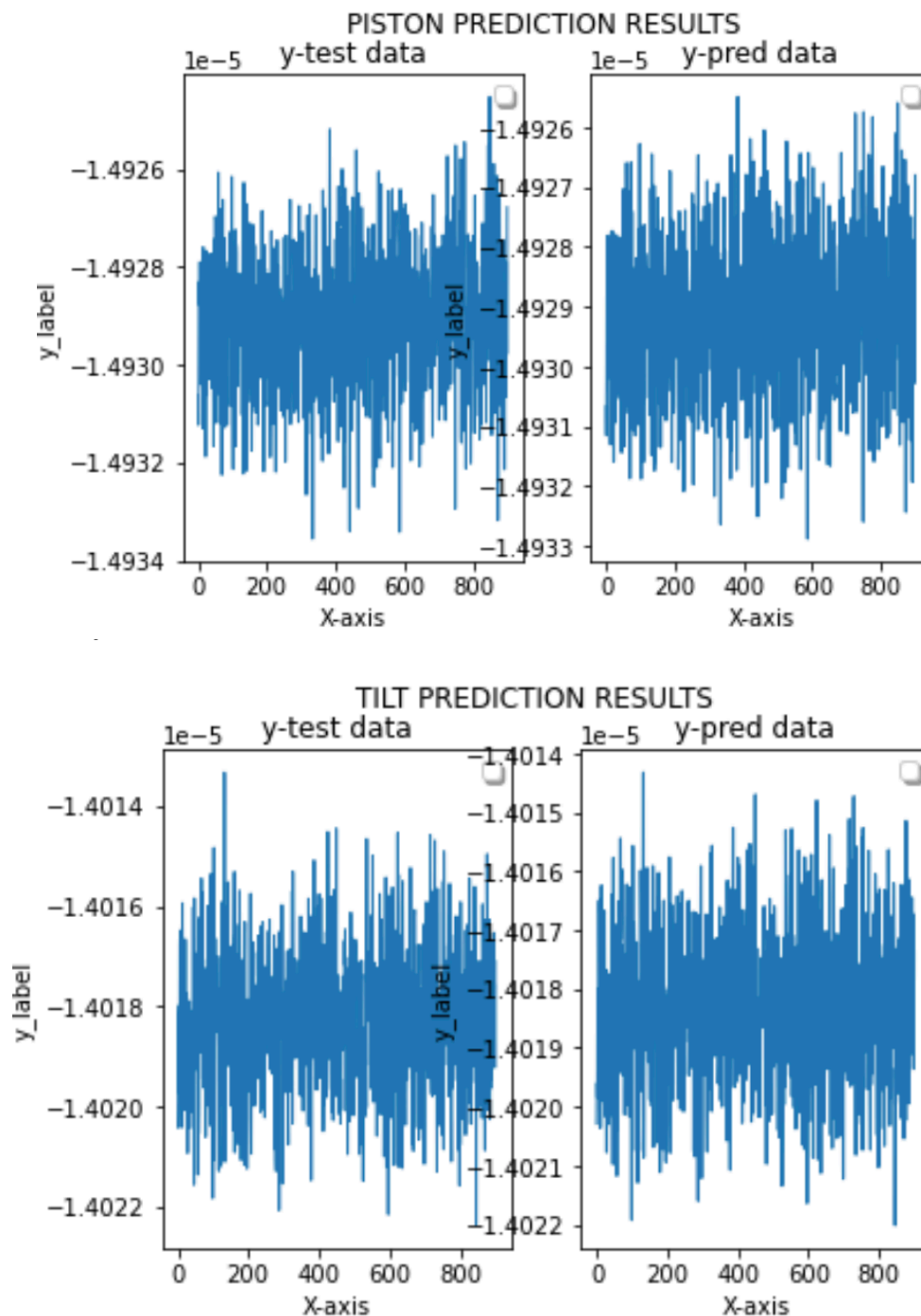
- [14] R. Landman, S. Y. Haffert, V. M. Radhakrishnan, and C. U. Keller "Self-optimizing adaptive optics control with reinforcement learning", Proc. SPIE 11448, Adaptive Optics Systems VII, 1144849 (13 de diciembre 2020); <https://doi.org/10.1117/12.2560053>
- [15] B. Pou, E. Quiñones, D. Gratadour, and M. Martin "Denoising wavefront sensor images with deep neural networks", Proc. SPIE 11448, Adaptive Optics Systems VII, 114484J (13 de diciembre); <https://doi.org/10.1117/12.2576242>
- [16] David Thomas, Joshua Meyers, and Steven M. Kahn "Wide-field wavefront sensing with convolutional neural networks and ordinary least squares", Proc. SPIE 11448, Adaptive Optics Systems VII, 114484H (13 de diciembre 2020); <https://doi.org/10.1117/12.2576020>
- [17] Maxime Quesnel, Gilles Orban de Xivry, Gilles Louppe, and Olivier Absil "Deep learning-based focal plane wavefront sensing for classical and coronagraphic imaging", Proc. SPIE 11448, Adaptive Optics Systems VII, 114481G (13 de diciembre 2020); <https://doi.org/10.1117/12.2562456>
- [18] Afshine, A. (n.d.). CS 230 - Recurrent Neural Networks Cheatsheet. Stanford Online, from <https://stanford.edu/%7Eshervine/teaching/cs-230/cheatsheet-recurrent-neural-networks>
- [19] S. Raj, J., & Ananthi J, V. (2019). RECURRENT NEURAL NETWORKS AND NONLINEAR PREDICTION IN SUPPORT VECTOR MACHINES. Journal of Soft Computing Paradigm, 2019(1), 33–40. <https://doi.org/10.36548/jscp.2019.1.004>
- [20] Staudemeyer, R. C. (Recuperado el 12 de septiembre del 2021). Understanding LSTM - a tutorial into Long Short-Term Memory Recurrent Neural Networks. ArXiv.Org. <https://arxiv.org/abs/1909.09586>
- [21] Liu, X., Morris, T., & Saunter, C. (2019). Using Long Short-Term Memory for Wavefront Prediction in Adaptive Optics. Artificial Neural Networks and Machine Learning – ICANN 2019: Text and Time Series, 537–542. https://doi.org/10.1007/978-3-030-30490-4_43
- [22] Chen, Y. (2020). LSTM recurrent neural network prediction algorithm based on Zernike modal coefficients. Optik, 203, 163796. <https://doi.org/10.1016/j.ijleo.2019.163796>
- [23] Chollet, F., & others. (2015). Keras. GitHub. Recuperado el 4 de mayo del 2021, en <https://github.com/fchollet/keras>
- [24] Rivera Ortega, A. U. (2018). Simple setup for wavelength estimation by using the Airy disk with didactical purposes. Optics Education and Outreach V. Published. <https://doi.org/10.1117/12.2306652>
- [25] Soria Hernández, E., López López, R., Oscoz Abad, A., & Colodro Conde, C. (2020). ALIOLI: presentation and first steps. Adaptive Optics Systems VII. Published. <https://doi.org/10.1117/12.2562277>
- [26] Taghinia, P., Muruganandan, V. A., Clare, R. M., & Weddell, S. J. (2020). A Wavefront Sensorless Tip/Tilt Removal method for Correcting Astronomical Images. 2020 35th International Conference on Image and Vision Computing New Zealand (IVCNZ). Published. <https://doi.org/10.1109/ivcnz51579.2020.9290688>

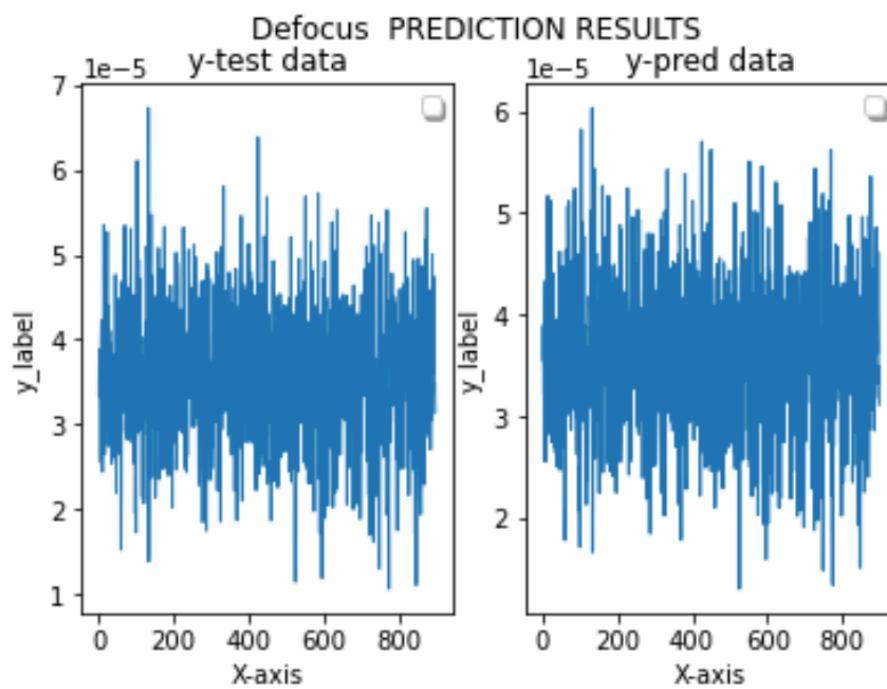
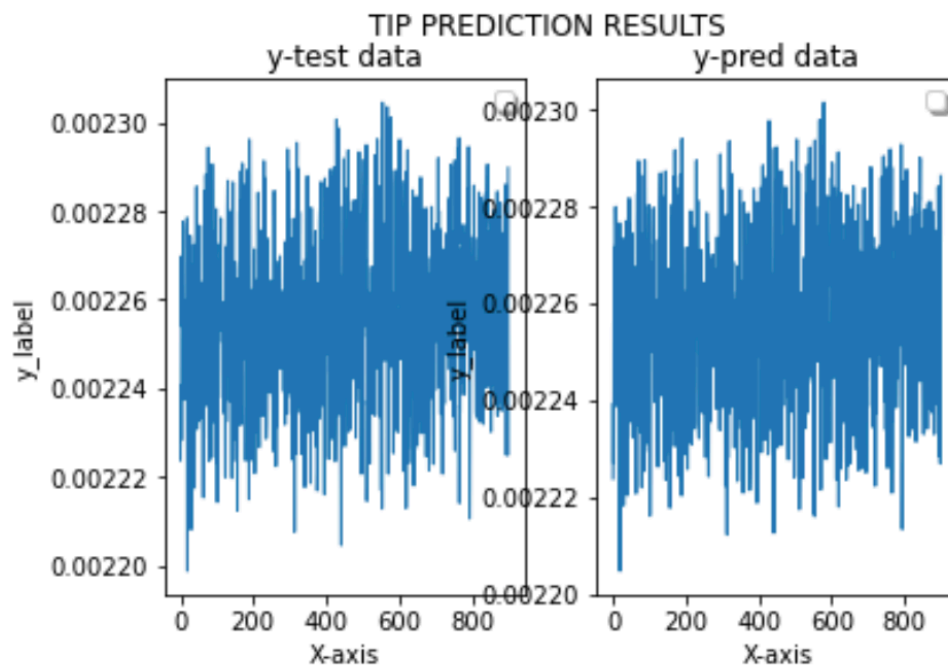
- [27] Liu, J., Muruganandan, V. A., Clare, R., Trujillo, M. C. R., & Weddell, S. J. (2020). A Tip-Tilt Mirror Control System for Partial Image Correction at UC Mount John Observatory. 2020 35th International Conference on Image and Vision Computing New Zealand (IVCNZ). Published. <https://doi.org/10.1109/ivcnz51579.2020.9290543>
- [28] Brownlee, J. (17 de enero del 2021) How to Develop a Bidirectional LSTM For Sequence Classification in Python with Keras. Machine Learning Mastery. <https://machinelearningmastery.com/develop-bidirectional-lstm-sequence-classification-python-keras/>
- [29] Great Learning Team. (16 de enero 2021). Mean Squared Error - Explained | What is Mean Square Error? GreatLearning Blog: Free Resources What Matters to Shape Your Career! <https://www.mygreatlearning.com/blog/mean-square-error-explained/>
- [30] Weddell, S.J., "Optical wavefront prediction with reservoir computing,"Ph.D. thesis (Department of Electrical & Computer Engineering, University of Canterbury, 2010. <http://dx.doi.org/10.26021/1679>
- [31] Weddell S.J. and Webb RY. (2008) Reservoir Computing for Prediction of the Spatially-Variant Point Spread Function. IEEE Journal of Selected Topics in Signal Processing 2(5): 624-634. <http://dx.doi.org/10.1109/JSTSP.2008.2004218>
- [32] Weddell S.J. and Bones P.J. (2018) Wavefront prediction with reservoir computing for minimizing the effects of angular anisoplanatism. Applied Optics 57(25): 7140-7151. <http://dx.doi.org/10.1364/AO.57.007140>

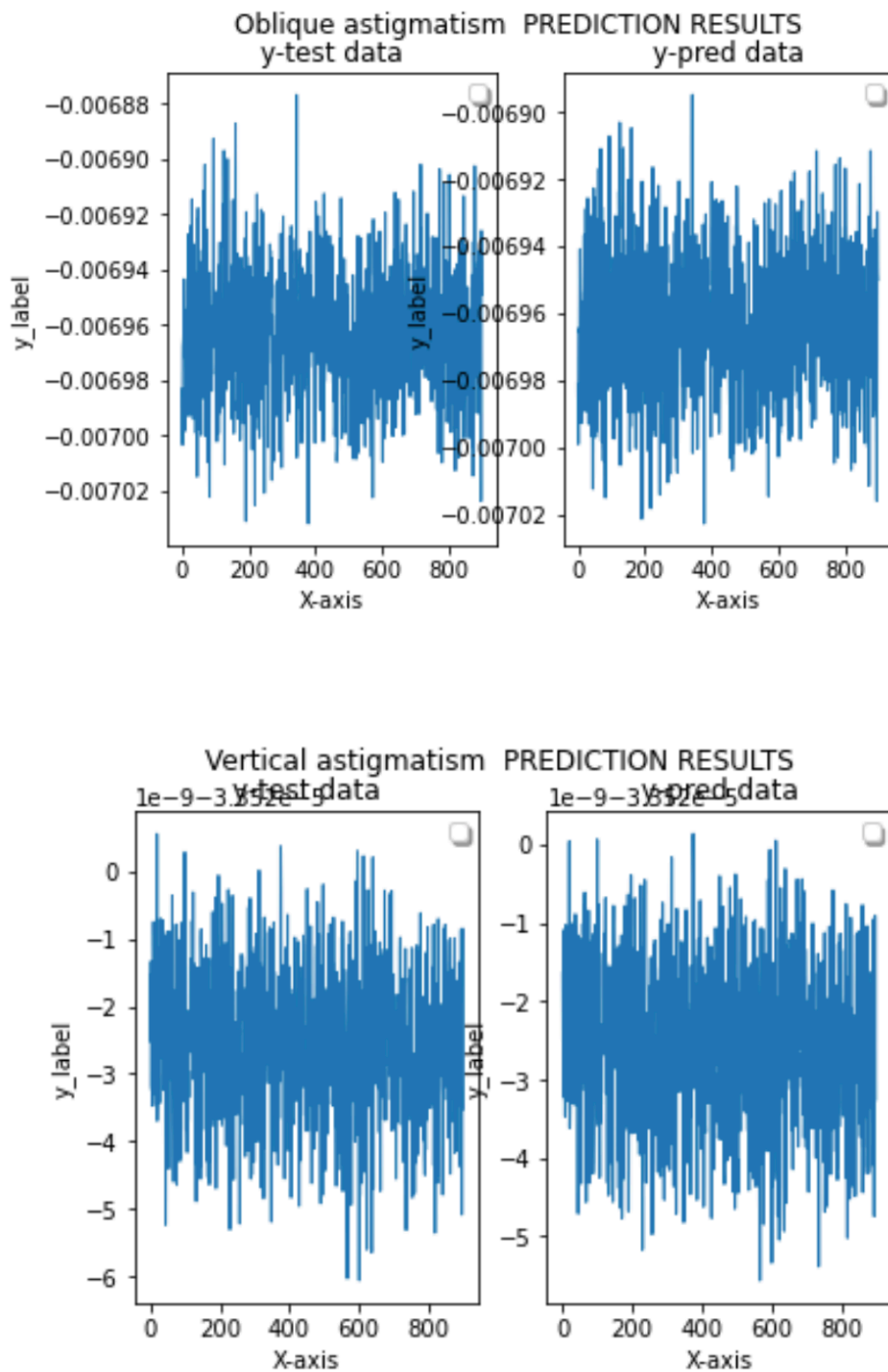
Anexos

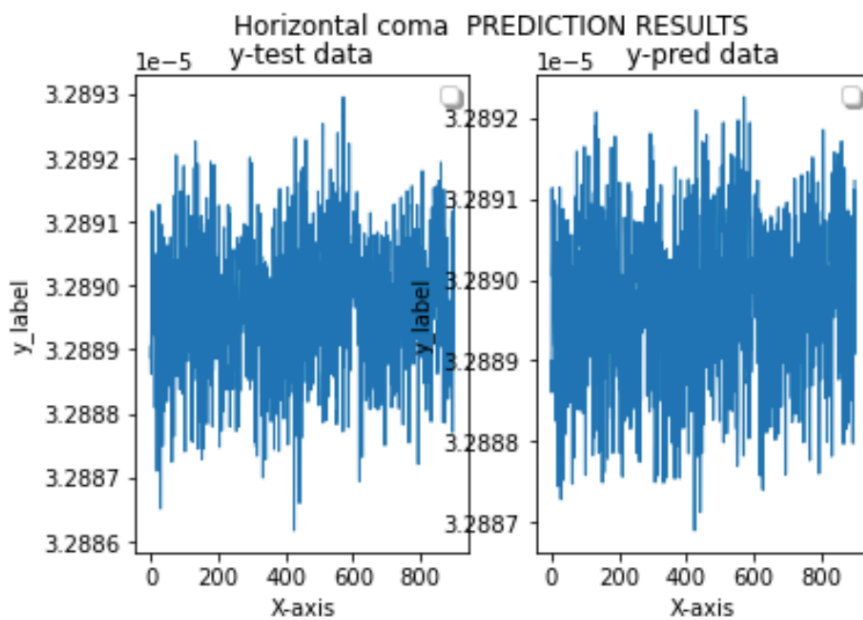
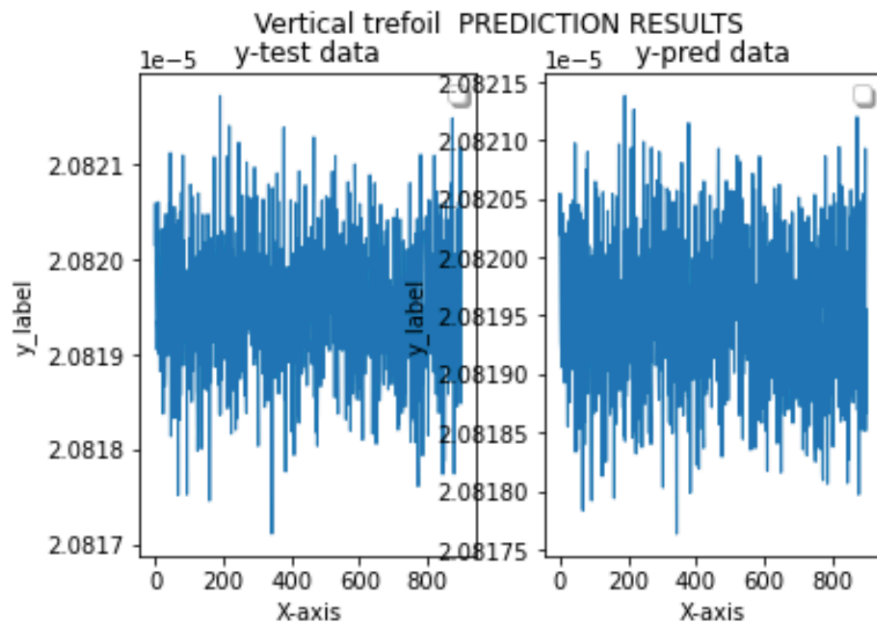
Anexo I. LSTM para AO: resultados

Nota: solo se van a representar los nueve primeros, por extensión. Se compararán, modo por modo, la aberración real con la predicha.









Anexo II. Artículo de investigación

AI4AO: The Inevitable future to see the stars

M^aCruz Ramírez Trujillo^{1,*}, Alberto Fernández Isabel¹, Stephen J. Weddell², Roberto López López³, Esther Soria Hernández³, Alejandro Oscoz Abad³



¹Universidad Internacional de la Rioja, Logroño (Spain)

²Department of Electrical and Computer Engineering; University of Canterbury, Christchurch (New Zealand)

³Instituto de Astrofísica de Canarias (IAC): La Laguna, Tenerife (Spain)

*Corresponding Author: mariadelacruz.ramirez071@comunidadunir.net

June 2021

ABSTRACT

A novel, Machine Learning method is presented to solve control-specific problems related to Adaptive Optics techniques, as well as an evidence of its use against other well-known techniques of adaptive control. Several methods are discussed for solving the presented problem [aberration correction in real time], from which the use of LSTM+RNN is highly recommended and defended in this report. The LSTM+RNN model has the capacity of calculating not just the first Zernike models but the entire range, at an optimal calculus speed of less than one millisecond per matrix, demonstrating its potency against more traditional opto-mechanical controllers. Plus, the LSTM+RNN model has showed improved results over time, overall improved processor performance and optimized calculus of a wavefront, Zernike based model. Furthermore, several improvements are discussed at the end of the report, where Artificial Intelligence methods are the key to new, improved Adaptive Optics.

KEYWORDS

Adaptative Control,
Adaptive Optics,
Machine Learning

I. INTRODUCTION

An Adaptive Optics system can correct and detect a distorted wavefront in real time, which improves the overall acquisition process. However, the usage of sensors and calculus for this goal delays the system, reducing the overall performance. The most common opto-mechanical system for an Adaptive Optics method consists of: (1) a Wavefront Sensor, (2) a controller for the correction loop (assuming a closed loop system) (3) the scientific camera, for e.g. an EMCCD Camera, (4) and the wavefront corrector, in this case, a Deformable Mirror (DM). In this paper a novel, Machine Learning based method is discussed. The main goal of the presented methodology is to replace the correction system, aka a commonly used PID corrector, for an Artificial Intelligence algorithm. The designed ML model will be used in a current AO group, called “ALIOLI” [5], state-of-the-art project from the Astrophysics Institute of the Canary Islands. Also, the CdA group of the University of Canterbury [6] is participant of the presented paper and project.

The proposed methodology consists of an improved, LSTM model that can solve the entire Zernike node range within milliseconds. This part will be discussed on section no.3. and section no.4. In section no.5, the results from the experiment of the presented LSTM model will be discussed.

II. STATE OF THE ART

For the correct development of the presented methodology, a former background research has been necessary. As AI applied to AO is concerned, several papers present novel methods that can discuss and prove the improvement of using these technologies in AO systems. On the system described by [13], a RNN model is designed for modelling the reconstruction of an upcoming wavefront. However, it doesn't imply that further control is applied to the system. In the search for more suitable

methodologies, the paper presented in [14] offers an interesting change of events: the usage of Reinforcement Learning in AO. This RL algorithm is therefore applied to a RNN as the main controller of the correction loop. This unique method, however, only offers the possibility of correcting the first two Zernike nodes, limiting its potential to Tip and Tilt correction. Deep Neural Networks are also being used in AO, while the acquisition process is taking place [15].

Discussing now the utility of Recurrent Neural Networks, [18] suggests that such models are ideally applied over sequential data, time dependent systems, where high performance from the main CPU is required. This notion of work is essential for an ideal AO operation, so further investigation was required on the subject.

On [21] an ANN-LSTM model is deployed over an AO system. In this case, there was no correction being made while the system was acquiring data, but they concentrated on obtaining an entire representation of the aberrated wavefront.

Furthermore, [22] presents an improved model based in RNN technology applied to the purpose that is being followed in this paper.

Considering the presented knowledge over AI and AO state-of-the-art methods, hence a novel LSTM method, python and Keras based model is presented. Acknowledging the efficiency and potential of Keras involving improved GPU usage, pre-trained networks and, more importantly, great documentation and ready-to-model networks, it was decided to work under this background.

III. OBJECTIVES AND METHODOLOGY

The principal objective of this work is to obtain a modern version of LSTM to work over a typical AO bench. For this purpose, a simulated library is proportionated by [5], and further support about the engineering proceedings is given by [6].

This LSTM model will work as a Slopes-to-Zernike converter, adapted to work primarily with a Shack-Hartman type Wavefront sensor. Further edits can be done easily, thanks to the mentioned library proportionated to simulate an AO bench, so other types of sensors can also work with this system. Furthermore, it should be capable of calculating a wide range of Zernike nodes. The calculus of this modes should not compromise the potential of the network, nor worsen the overall delay of the AO system. In any cases, it is expected to rather improve highly. To summarize this section, we need an automatized process of sequence-to-sequence conversion between slopes and Zernike modes.

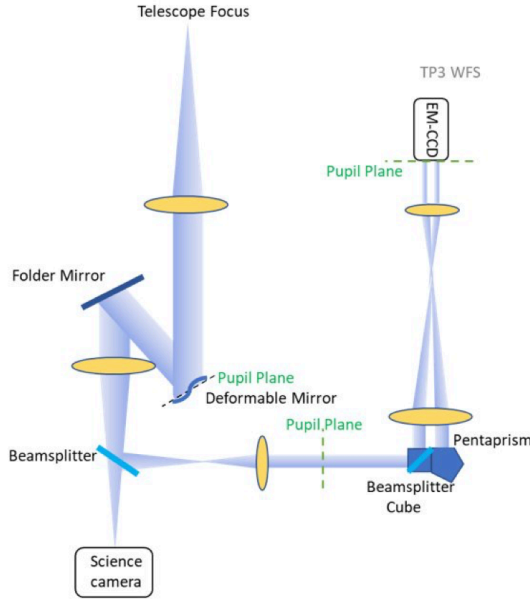


Fig. 1. Adaptive Optics bench used by ALIOLI project. Source: [5]

Regarding more specific objectives, the model must be able to work with large amount of data. Due we are evaluating sequential data, that is, a vast number of images over a short period of time, the presented model must be capable of working with the dataset and still meet the requirements. Under the AO, python-based library restrictions, we can work with the commonly used image size of 256x256 pixels. From this size we will extract 178 points, whereas each point represents a slope. Each slope will be calculated from an image of reference. A slope represents the deviation of a vector position from its initial point. Without no more previous data adjustment being necessary, the dataset will be ready to function as an input of the LSTM model. As a good predictor of a wavefront shape, this model has to be also capable of predicting, at least, more than 20 Zernike nodes. Summarizing the details, the LSTM must work with input sizes over 178 features and give a sequence of more than 20 points as a result. This may look rather complicated, but the capability that LSTM shows of remembering previous states gives a great help for optimizing the computational cost, also giving a more exact prediction.

The approach being taken for solving the presented problem is to study and discuss several machine learning methods, comparing then against the proposed LSTM model. The test will be taking at same conditions for each one of the models. These models will consist of: (1) the proposed LSTM, (2) a different approach of

LSTM, adding a PID simulated behavior and (3), a simpler ANN, linear regression model. By comparing their results over a period of over 3000 iterations, the expected results will be of a good improvement in terms of accuracy and convergence time over less than 50 epochs. For this purpose, complementary tables and graphs will be shown later on this paper, under section number 5.

TABLE I

WAVEFRONT RMS ERROR FROM LGS AO PROJECT. REFERENCE: [7]

Wavefront Error Term	NM RMS
Atmospheric Fitting	122
Telescope Fitting	66
Science Camera	110
Wavefront Bandwidth	182
Wavefront measurement	216
LGS Focus Error	70
Focus Anisoplanatism	181
LGS high order error	80
Calibration Errors	30
Miscellaneous	73
Total High Order Wavefront Error	402

Considering the results given by [7], it shows perfectly the error that must be improved with the given AI system. Since the performance of the LSTM model depends mainly on the GPU being used to train it, we can resume its timing error depending on its training optimization.

IV. CONTRIBUTION

The LSTM model presented in this paper contributes utterly in AO and AI methodologies, throwing on a new perspective of using powerful frameworks such as Tensorflow and Keras. Traditional neural network designing methods are proving to rather slow the overall project development, since large amount of time is being spent on coding.

Further contribution has been made by changing from traditional neural network development tools, as Matlab, to Python. We can offer a “simpler” network in appearance, with no more than two sequential layers as main structure. Powerful enough to use its technology sequence-to-sequence, this methodology is capable of converting the 178 slope points mentioned in section III to 23 Zernike nodes in less than one millisecond per matrix. This result has surpassed all expectations, since not that big of a change was expected. Further evaluation will be granted and explained in section V.

With this network, a high-performance AO correction loop is achieved. For correcting the Deformable Mirror in real time, the only thing necessary between the science camera and the wavefront sensor is a GPU.

V. EVALUATION AND RESULTS

In this section will be discussed the results from the evaluation of the LSTM model against a timeseries aberration. For this purpose, the procedure has been marked with the following characteristics: (1) light wavelength on the range of visible light ($700e^{-9}$ nm), (2) a medium-small telescope diameter is assumed, (3) zero-magnitude flux is set to be $3.9e^{10}$.

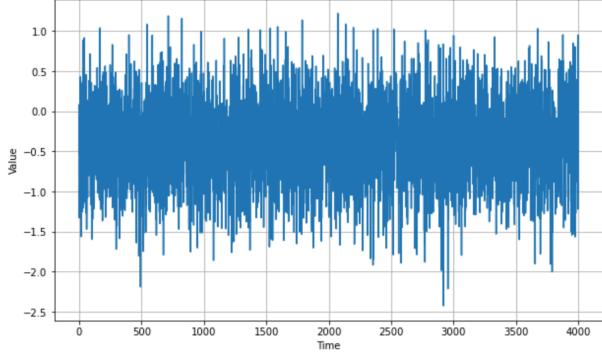


Fig 2. Time series evolution for tilt aberration

Before feeding the data to the presented networks, it is necessary to use a standard scaler over the entire training set. The “label” data that is used for both training and evaluation is quite small: the higher the Zernike mode, the smaller its value. So, in order to get better results, it is necessary to standardize the data.

After defining the models, the optimization process and training parameters are shared over the three ML models. After doing an extensive grid search, the best results were obtained with an Adam solver. The metrics used during the optimization process are the mean squared error (MSE), the mean absolute error (MAE) and the accuracy. Loss will be computed using the MSE value.

Model no.1: Simpler LSTM

Layer (type)	Output Shape	Param #
lstm_3 (LSTM)	(None, 1, 178)	254184
dense_2 (Dense)	(None, 1, 23)	4117
Total params: 258,301		
Trainable params: 258,301		
Non-trainable params: 0		

Fig 3. Simpler LSTM model overview

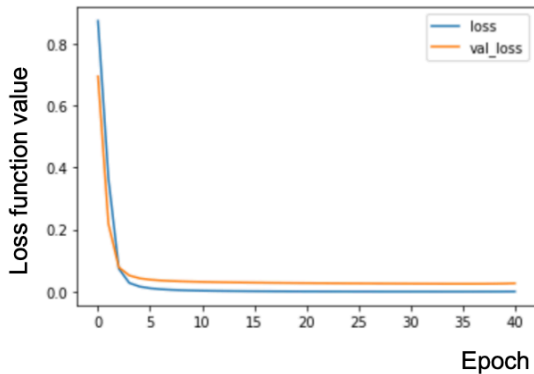


Fig 4. Loss function over the evaluated epochs for Simpler LSTM

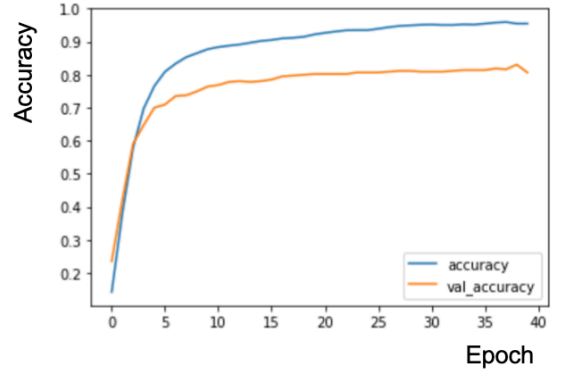


Fig 5. Training accuracy against validation accuracy over training epoch – Simpler LSTM

TABLE II
LSTM EVALUATION RESULTS

Parameter	VALUE
Validation set accuracy – peak value	0.83095
Training accuracy – peak value	0.95952
Loss – Validation set	0.025
Loss – Training set	0.0003218
Speed performance over a matrix	0.06 ms / matrix
Training time	10.46 seconds
RMSE value	6.069e-06

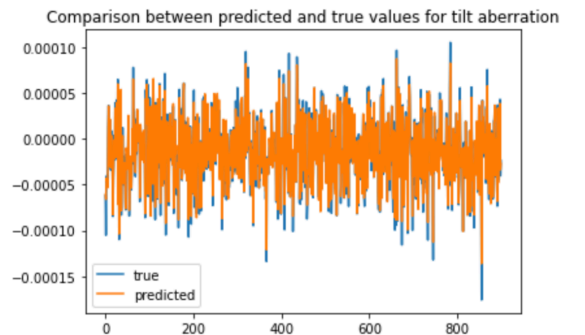


Fig 5. Simulated data against predicted data –Simpler LSTM

Model no.2: LSTM with PID behavior

This model implements the same sequential LSTM model presented in figure number 3, but with an early step added after the standard scaler has been applied. The data is easily model as a PID following the formula presented in figure 6.

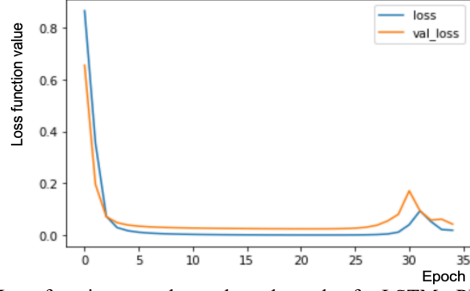


Fig 6. Loss function over the evaluated epochs for LSTM+PID

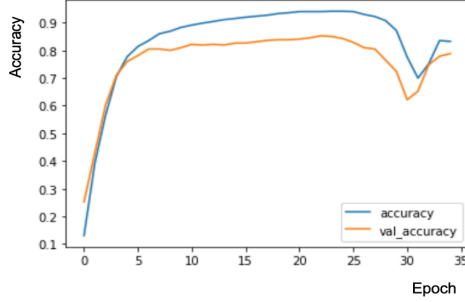


Fig 7. Training accuracy against validation accuracy over training epoch - LSTM+PID

TABLE III
LSTM PLUS PID EVALUATION RESULTS

Parameter	VALUE
Validation set accuracy – peak value	0.833
Training accuracy – peak value	0.93
Loss – Validation set	0.028
Loss – Training set	0.00155
Speed performance over a matrix	0.06 ms / matrix
Training time	8.115 seconds
RMSE value	4.36e-04

Model no.3: Linear Regression over ANN

This simpler ANN model follows a typical Linear Regression model. No further data

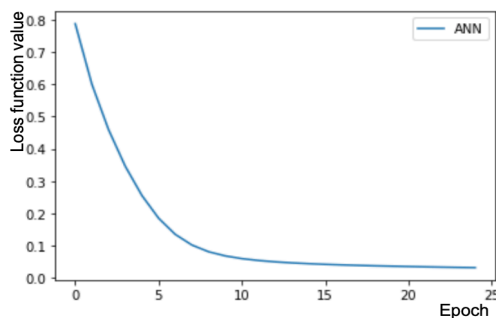


Fig 8. Loss function over epoch for the ANN model

TABLE II
ANN EVALUATION RESULTS

Parameter	VALUE
Score	0.95627
Loss value	0.0311
Speed performance over a matrix	0.05 ms / matrix
Training time	0.3026 seconds
RMSE value	4.38e-04

COMPARISON BETWEEN MODELS

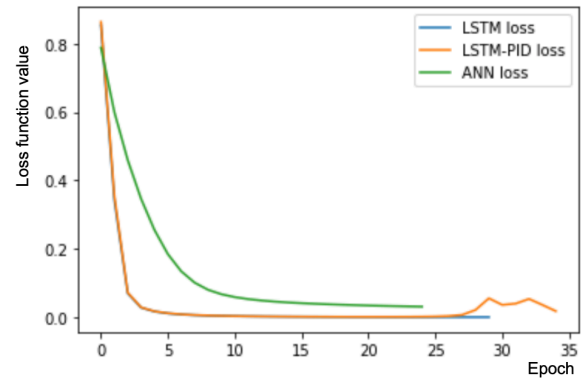


Fig 9. Cost function comparison between models

VI. DISCUSSION

In this section, the results presented in section V will be discussed.

Considering the objectives described in section III, it's right to declare that all three models work for the given purpose. However, accuracy and stableness are important factors when working against atmosphere. Given this notion, ANN is depreciated against the proposed LSTM models. If precision was not such a decisive factor, ANN would be the solution to go, but it has more problems attached to its model. The main problem that an ANN model presents for predicting aberration over time is that it won't be able to recognize a sequential pattern. If atmosphere changes much or not, it will have problems adapting to new situations.

The discussion is between using a modelled PID inside the LSTM and a much simpler LSTM. This approach was taken under the notion that the system could behave as the existing PID controller. It has nice results over the sequence, but the model tends to be quite unstable whenever higher epochs are approached. This can be seen in figures 6 and 7.

The much simpler LSTM converges faster, with more steadiness over time and higher accuracy is reached. Therefore, the LSTM model is the ideal behavior that compromises optimal cost, optimal accuracy at a high calculus rate.

VII. CONCLUSION

The presented methodology shows that an upcoming aberrated wavefront can be translated as a sequence-to-sequence case under a LSTM model. Thanks to this adaptation, great success was reached, without compromising the project requirements at all. The results have exceeded the expectations, and worth fighting for. Even if LSTM are primarily used for Natural Language Processing, the results give an intuition that we can expect a similar behavior when translating from slopes to Zernike modes.

A system capable of translating slopes to a modeled wavefront at the given rates are a notorious improvement for AO techniques, in general. This work could be improved if we introduce a CNN model within the LSTM, so it could receive the unprocessed images from the star right from the beginning. If the system proves capable of converting images into modeled wavefront, it would make unnecessary to keep a WFS at all. The goal that is being flighted for is to remove as much sensors as possible from the AO bench. At the end, it could be realistic to expect that only a science camera and a pre-trained network over a GPU would be the only elements conforming an AO bench. This would reduce the induced error from the several components conforming this system even more.

Artificial Intelligence methodologies are the future of Astrophysics. Its technology guarantees adaptability, accuracy, and even better automation over the existing methods.

REFERENCES

- [1] Friedlander, M. W. (n.d.). Astronomy - The techniques of astronomy. Encyclopedia Britannica. Retrieved April 26, 2021, from <https://www.britannica.com/science/astronomy/The-techniques-of-astronomy#ref910526>
- [2] Mackay, C. (n.d.). High-resolution imaging with large ground-based telescopes. Spie.Org. Retrieved May 4, 2021, from <https://spie.org/news/5408-high-resolution-imaging-with-large-ground-based-telescopes?SSO=1>
- [3] information@eso.org. (n.d.). Adaptive Optics. ESO. Retrieved May 4, 2021, from https://www.eso.org/public/teles-instr/technology/adaptive_optics/
- [4] Leigui de Oliveira, M., & Gabriel, J. P. S. (2018). The characterization of the Zernike modes at the focal plane for Extremely Large Telescope projects. Adaptive Optics Systems VI, 2–3. <https://doi.org/10.1117/12.2315537>
- [5] AOLI-Presentation. (n.d.). Copyright 2021 AOLI. All Rights Reserved. (<http://Research.Iac.Es/Proyecto/AOLI>). <http://research.iac.es/proyecto/AOLI/>
- [6] The University of Canterbury. UC Research Profile - University of Canterbury - New Zealand. <https://researchprofile.canterbury.ac.nz/Group.aspx?groupid=45>
- [7] Wizinowich, P. (2015). Adaptive optics in astronomy. Contemporary Physics, 56(4), 432–450. <https://doi.org/10.1080/00107514.2015.1041765>
- [8] Edmund Optics. (2021). Introduction to Adaptive Optics and Deformable Mirrors. Edmund Optics Inc. <https://www.edmundoptics.com/knowledge-center/application-notes/optics/introduction-to-adaptive-optics-and-deformable-mirrors/>
- [9] Colodro-Conde, C., Velasco, S., Fernández-Valdivia, J. J., López, R., Oscoz, A., Rebol, R., Femenía, B., King, D. L., Labadie, L., Mackay, C., Muthusubramanian, B., Pérez Garrido, A., Puga, M., Rodríguez-Coira, G., Rodríguez-Ramos, L. F., Rodríguez-Ramos, J. M., Toledo-Moreo, R., & Villó-Pérez, I. (2017). Laboratory and telescope demonstration of the TP3-WFS for the adaptive optics segment of AOLI. Monthly Notices of the Royal Astronomical Society, 467(3), 2855–2868. <https://doi.org/10.1093/mnras/stx262>
- [10] Colodro-Conde, C., Velasco, S., López, R., Oscoz, A., Martán-Hernando, Y., Rebol, R., Pérez-Garrido, A., Fernández-Valdivia, J. J., Labadie, L., Mackay, C., Puga, M., Rodríguez-Coira, G., Rodríguez-Ramos, L. F., & Rodríguez-Ramos, J. M. (2017). The TP3-WFS: a new guy in town. Proceedings of the Adaptive Optics for Extremely Large Telescopes 5, 1. <https://doi.org/10.26698/ao4elt5.0167>
- [11] Education, I. C. (2021, April 7). Recurrent Neural Networks. IBM. <https://www.ibm.com/cloud/learn/recurrent-neural-networks>
- [12] Search the world's largest collection of optics and photonics applied research. (n.d.). SPIE Digital Library. <https://www.spiedigitallibrary.org>
- [13] Pedro Escárate, Gustavo Rahmer, and Julian Christou "Neural network wavefront sensor for the Large Binocular Telescope Observatory", Proc. SPIE 11448, Adaptive Optics Systems VII, 114484F (13 December 2020); <https://doi.org/10.1117/12.2562406>
- [14] R. Landman, S. Y. Haffert, V. M. Radhakrishnan, and C. U. Keller "Self-optimizing adaptive optics control with reinforcement learning", Proc. SPIE 11448, Adaptive Optics Systems VII, 1144849 (13 December 2020); <https://doi.org/10.1117/12.2560053>
- [15] B. Pou, E. Quiñones, D. Gratadour, and M. Martin "Denoising wavefront sensor images with deep neural networks", Proc. SPIE 11448, Adaptive Optics Systems VII, 114484J (13 December 2020); <https://doi.org/10.1117/12.2576242>
- [16] David Thomas, Joshua Meyers, and Steven M. Kahn "Wide-field wavefront sensing with convolutional neural networks and ordinary least squares", Proc. SPIE 11448, Adaptive Optics Systems VII, 114484H (13 December 2020); <https://doi.org/10.1117/12.2576020>
- [17] Maxime Quesnel, Gilles Orban de Xivry, Gilles Louppe, and Olivier Absil "Deep learning-based focal plane wavefront sensing for classical and coronagraphic imaging", Proc. SPIE 11448, Adaptive Optics Systems VII, 114481G (13 December 2020); <https://doi.org/10.1117/12.2562456>
- [18] Afshine, A. (n.d.). CS 230 - Recurrent Neural Networks Cheatsheet. Stanford Online, from <https://stanford.edu/%7Eshervine/teaching/cs-230/cheatsheet-recurrent-neural-networks>
- [19] S. Raj, J., & Ananthi J, V. (2019). RECURRENT NEURAL NETWORKS AND NONLINEAR PREDICTION IN SUPPORT VECTOR MACHINES. Journal of Soft Computing Paradigm, 2019(1), 33–40. <https://doi.org/10.36548/jscp.2019.1.004>
- [20] Staudemeyer, R. C. (2019, September 12). Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks. ArXiv.Org. <https://arxiv.org/abs/1909.09586>
- [21] Liu, X., Morris, T., & Saunter, C. (2019). Using Long Short-Term Memory for Wavefront Prediction in Adaptive Optics. Artificial Neural Networks and Machine Learning – ICANN 2019: Text and Time Series, 537–542. https://doi.org/10.1007/978-3-030-30490-4_43
- [22] Chen, Y. (2020). LSTM recurrent neural network prediction algorithm based on Zernike modal coefficients. Optik, 203, 163796. <https://doi.org/10.1016/j.ijleo.2019.163796>
- [23] Chollet, F., & others. (2015). Keras. GitHub. Retrieved from <https://github.com/fchollet/keras>
- [24] Rivera Ortega, A. U. (2018). Simple setup for wavelength estimation by using the Airy disk with didactical purposes. Optics Education and Outreach V. Published. <https://doi.org/10.1117/12.2306652>
- [25] Soria Hernández, E., López López, R., Oscoz Abad, A., & Colodro Conde, C. (2020). ALIOLI: presentation and first steps. Adaptive Optics Systems VII. Published. <https://doi.org/10.1117/12.2562277>
- [26] Taghinia, P., Muruganandan, V. A., Clare, R. M., & Weddell, S. J. (2020). A Wavefront Sensorless Tip/Tilt Removal method for Correcting Astronomical Images. 2020 35th International Conference on Image and Vision Computing New Zealand (IVCNZ). Published. <https://doi.org/10.1109/ivcnz51579.2020.9290688>
- [27] Liu, J., Muruganandan, V. A., Clare, R., Trujillo, M. C. R., & Weddell, S. J. (2020). A Tip-Tilt Mirror Control System for Partial Image Correction at UC Mount John Observatory. 2020 35th International Conference on Image and Vision Computi x

New Zealand (IVCNZ). Published.
<https://doi.org/10.1109/ivcnz51579.2020.9290543>

[28] Brownlee, J. (2021, January 17). How to Develop a Bidirectional LSTM For Sequence Classification in Python with Keras. Machine Learning Mastery. <https://machinelearningmastery.com/develop-bidirectional-lstm-sequence-classification-python-keras/>

[29] Great Learning Team. (2021, June 16). Mean Squared Error - Explained | What is Mean Square Error? GreatLearning Blog: Free Resources What Matters to Shape Your Career! <https://www.mygreatlearning.com/blog/mean-square-error-explained/>

[30] Weddell, S.J., "Optical wavefront prediction with reservoir computing," Ph.D. thesis (Department of Electrical & Computer Engineering, University of Canterbury, 2010. <http://dx.doi.org/10.26021/1679>

[31] Weddell SJ. and Webb RY. (2008) Reservoir Computing for Prediction of the Spatially-Variant Point Spread Function. IEEE Journal of Selected Topics in Signal Processing 2(5): 624-634. <http://dx.doi.org/10.1109/JSTSP.2008.2004218>

[32] Weddell SJ. and Bones PJ. (2018) Wavefront prediction with reservoir computing for minimizing the effects of angular anisoplanatism. Applied Optics 57(25): 7140-7151. <http://dx.doi.org/10.1364/AO.57.007140>