



**UNIVERSITAT
JAUME·I**

UNIVERSITAT JAUME I

**ESCOLA SUPERIOR DE TECNOLOGIA I CIÈNCIES EXPERIMENTALS
GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES**

SEGUIDOR SOLAR DE DOS EJES

TRABAJO FIN DE GRADO

AUTOR

Sergio Balaguer Zueco

DIRECTORES

Enrique Belenguer Balaguer

José Manuel Portolés Flaj

SEGUIDOR SOLAR DE DOS EJES



Agradecimientos:

A mis tutores Enrique Belenguer Balaguer y José Manuel Portolés Flaj, por el soporte, ayuda y paciencia que han tenido.

Al equipo éBrickhouse. Por la inspiración y las experiencias vividas.

A Teresa Gallego Navarro y a Raúl Marín Prades, por su colaboración en el proceso de materialización de las prácticas asociadas al TFG.

Onda, 09/11/2016

1. MEMORIA	9
1.1 OBJETO	9
1.2 ALCANCE	10
1.3 ANTECEDENTES	11
1.4 NORMAS Y REFERENCIAS	12
1.4.1 DISPOSICIONES LEGALES Y NORMAS APLICADAS	12
1.4.2 PROGRAMAS DE CÁLCULO	13
1.4.3 PLAN DE GESTIÓN DE LA CALIDAD APLICADO DURANTE LA REDACCIÓN DEL PROYECTO	14
1.4.4 BIBLIOGRAFÍA	15
1.5 DEFINICIONES Y ABREVIATURAS	15
1.6 REQUISITOS DE DISEÑO	22
1.6.1. DATOS DE PARTIDA	22
1.6.2. EMPLAZAMIENTO	24
1.7 ANÁLISIS DE SOLUCIONES	26
1.7.1. SEGUIMIENTO	26
1.7.2. ESTRUCTURA.	31
1.8 RESULTADOS FINALES	32
1.8.1. MECANISMO	32
1.8.2. ESTRUCTURA	33
1.8.3. PLACA ANCLAJE	34
1.8.4. CIMENTACIÓN.	35
1.8.5. SELECCIÓN DE ACUADORES Y TRANSMISIÓN.	35
MOVIMIENTO DE ELEVACIÓN:	35
MOVIMIENTO DE ROTACIÓN	35
1.8.9. ENUMERACIÓN DE CUESTIONES PENDIENTES DE RESOLVER	36
1.9 PLANIFICACIÓN	36
1.9.1. DIAGRAMA DE GANTT	37
1.9.2. DIAGRAMA DE REDES Y CAMINO CRÍTICO	38
1.10 ORDEN DE PRIORIDAD ENTRE LOS DOCUMENTOS	39

2. ANEJOS A LA MEMORIA	41
2.1. ESTRATEGIA DE SEGUIMIENTO Y RESOLUCIÓN	41
2.2. DESCRIPCIÓN DEL MECANISMO	43
2.3. DESCRIPCIÓN DE ACTUADORES	45
2.4. ANÁLISIS DEL MECANISMO DE ELEVACIÓN	47
2.4.1. ESTUDIO DE LIMITACIONES FÍSICAS EN FUNCIÓN DE LOS PARÁMETROS DEL MECANISMO:	49
2.4.2. SELECCIÓN DE LOS PARÁMETROS DEL MECANISMO	53
2.5. PARÁMETROS DE LA TRANSMISIÓN.	55
2.5.1. MECANISMO DE ELEVACIÓN.	56
2.5.2. MECANISMO DE ROTACIÓN	57
2.6. CONTROL DEL MECANISMO	58
2.6.1. MICROCONTROLADOR	58
2.6.2. ENTORNO DE PROGRAMACIÓN	60
2.6.3. LIBRERÍAS Y FUNCIONES <i>BUILT IN</i> UTILIZADAS	62
2.7. DESCRIPCIÓN DEL PROGRAMA	64
2.7.1. CONSTANTES NECESARIAS:	64
2.7.2. VARIABLES NECESARIAS:	65
2.7.3. FUNCIONES NECESARIAS:	66
2.7.4. RESUMEN DEL PROGRAMA.	67
CÁLCULO DE LOS ÁNGULOS DE SALIDA DEL SOL	71
2.7.5. EXPLICACIÓN DE LAS FUNCIONES UTILIZADAS.	77
FUNCIONES PARA LA GESTIÓN DEL TIEMPO	77
FUNCIONES TRIGONOMÉTRICAS.	81
FUNCIONES DE CÁLCULO DE PARÁMETROS SOLARES	83
ACTIVACIÓN DE LOS MOTORES PASO A PASO	84
2.7.6. FUNCIONES DE CONTROL DE LOS STEPPERS	88
MOVIMIENTO DE ELEVACIÓN:	88
MOVIMIENTO DE ROTACIÓN:	90
PARO DE LOS MOTORES	90
2.7.7. CONTROL DEL TIEMPO	91
2.7.8. PROGRAMA COMPLETO	92

2.7.9. INICIALIZACIÓN _____	106
CONSTANTES _____	106
VARIABLES _____	106
2.7.10. CONEXIONADO _____	107
2.8. SELECCIÓN DE MATERIALES ESTRUCTURALES _____	108
2.9. CÁLCULOS ESTRUCTURALES. _____	112
2.9.1. PLANTEAMIENTO _____	112
2.9.2. RESULTADOS DEL CÁLCULO _____	121
DATOS DE OBRA: _____	121
COMBINACIONES DE ESFUERZOS _____	124
COMPROBACIONES ELU. FLECHA _____	128
COMPROBACIONES ELU. EJEMPLO UNA BARRA. _____	129
2.9.3. CIMENTACIÓN _____	138
DESCRIPCIÓN _____	138
MEDICIÓN _____	138
COMPROBACIÓN _____	139
2.9.4. FIJACIÓN DE LAS PLACAS _____	141
2.10. CÁLCULO DE LOS ACTUADORES _____	142
2.10.1. MOTOR DE ELEVACIÓN _____	142
RESULTADO _____	145
2.10.2. MOTOR DE ROTACIÓN. _____	145
RESULTADO _____	148

3. PLANOS	149
4. PLIEGO DE CONDICIONES	151
4.1. PANELES SOLARES.	151
DESCRIPCIÓN.	151
ESPECIFICACIONES.	151
4.2. MICROCONTROLADOR.	151
DESCRIPCIÓN	151
ESPECIFICACIONES.	152
4.3. ESTRUCTURA Y CIMENTACIÓN.	152
DESCRIPCIÓN	152
ESPECIFICACIONES	152
5. PRESUPUESTO	153
5.1. PARTE ESTRUCTURAL.	153
5.2. PARTE ELECTRÓNICA	154
5.3.PRESUPUESTO TOTAL	154

1. Memoria

1.1 Objeto

El objetivo del presente proyecto es llevar a cabo el diseño de un seguidor solar de dos ejes que sea capaz de orientar hasta cuatro placas solares.

Un seguidor solar es un mecanismo de soporte para placas solares, cuyo principal objetivo es situar las placas en todo momento en una posición óptima para la generación de la máxima energía posible.

Hay dos tipos de seguidores:

Seguidores de un eje

Tienen un solo un grado de libertad. Se caracterizan por realizar un seguimiento parcial de la trayectoria del Sol. Pueden ser de tipo polar, horizontal o acimutal, dependiendo de la orientación y posición del eje móvil.

Seguidores de dos ejes.

Tienen dos grados de libertad. Consiguen realizar un seguimiento completo de la trayectoria solar, con lo cual se obtiene la producción de energía máxima.

Nuestro seguidor será de este segundo tipo. Al tener dos ejes móviles, tendremos dos grados de libertad que controlar, lo cual se corresponde con dos actuadores como mínimo. Así mismo, buscaremos que consuma la mínima energía posible. Para ello:

Utilizaremos mecanismos con autorretención, para no tener que suministrar energía para mantener la posición

Controlar la posición mediante las ecuaciones que describen el movimiento del Sol, para evitar el consumo de sensores adicionales.

Realizar un movimiento discontinuo, para minimizar el consumo eléctrico

Además, se intentará que el seguidor sea fácil de construir para abaratar costes y hacer accesible su utilización a todo tipo de públicos. E incluso que en la medida de lo posible una persona con un mínimo de formación sea capaz de construirse para uso personal.

1. Memoria

1.2 Alcance

Este proyecto está orientado para todas aquellas aplicaciones en las que se quiera maximizar la captación de energía solar. Ya sea en ámbito público o privado; para ciudadanos particulares o empresas.

Como puedan ser aplicaciones en las que el número de paneles a instalar haya de ser bajo y se desee obtenerles el máximo rendimiento. Como por ejemplo:

Estaciones aisladas de bombeo:

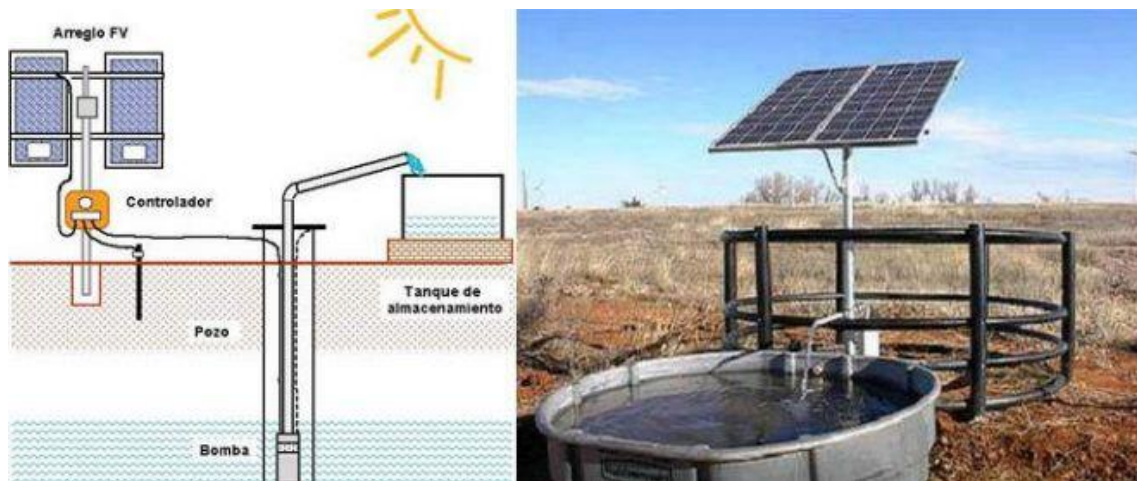


Fig. 1 Kit suministrado por la empresa AutoSolar.

Autoconsumo en construcciones aisladas de red:

Dos instalaciones realizadas por la empresa ALCAZAREN SOSTENIBLE



a)



b)

Fig. 2 Dos instalaciones realizadas por la empresa ALCAZAREN SOSTENIBLE

1.3 Antecedentes

Cálculos previos y justificación:

En los últimos años, el precio de las placas solares ha bajado considerablemente.

Entre el 2008 y el 2011 el precio del kilo de Silicio bajó de 450\$ a menos de 27\$. Además cada vez los procesos de fabricación son mejores, consiguiendo paneles más eficientes, que consiguen rendimientos muy buenos aún con condiciones meteorológicas poco favorables.

Y teniendo en cuenta la entrada de China al sector de fabricación de placas fotovoltaicas, actualmente tenemos unos precios que rondan los 0,80€ por vatio.

En lo que respecta a la producción de energía, instalando sistemas de seguimiento podemos aumentar la producción hasta en un 40%. No obstante, el espacio necesario para la instalación es entorno a un 50% mayor.

Por eso, la mejor opción para instalar un seguidor será cuando tengamos muy poco espacio para colocar las placas o pocas placas a instalar. Y por tanto, necesitemos sacar el máximo rendimiento a la instalación.

Si nos encontramos en las condiciones que acabamos de describir, entonces instalar un seguidor de estas características será una de las mejores opciones. Y sin duda, la más eficaz.

No obstante para instalaciones en las que tengamos mayor superficie y podamos instalar tantas placas como deseemos, habrá que hacer un estudio previo para valorar las opciones y obtener una solución óptima.

En dicha situación, los principales factores a tener en cuenta serán:

Con seguimiento	Sin seguimiento
Precio del sistema de seguimiento + mantenimiento	Caben mas placas por unidad de superficie
Aumento de la superficie necesaria	Producción unitaria menor
Mayor producción unitaria	
Precio de cada Placa	
Superficie disponible	
Variación de las condiciones de irradiancia a lo largo del año	

1.4 Normas y Referencias

1.4.1 Disposiciones legales y normas aplicadas

Para llevar a cabo los cálculos estructurales del presente proyecto, se han utilizado:

El Documento Básico sobre Seguridad Estructural para el cálculo de acciones en la edificación, del Código Técnico de Edificación (CTE DEB SE AE)

El Documento Básico sobre Seguridad Estructural para el cálculo y dimensionado de Estructuras de Acero (CTE DB SE - A)

El Documento EHE 98 CTE de la Instrucción Española del Hormigón Estructural, para el cálculo de la zapata y el encofrado necesario para el anclaje del seguidor.

Para el cálculo del presupuesto de la parte estructural se ha utilizado la base de datos: <http://itec.es/noubedec.e/bedec.aspx>

Con las siguientes referencias:

Referencia	Aspecto cubierto
E441F135	Acero S275J0H según UNE-EN 10210-1, para pilares formados por pieza simple, en perfiles huecos laminados en caliente serie redondo, cuadrado y rectangular, trabajado en taller y con una capa de imprimación antioxidante, colocado en obra con soldadura
B0CHL50A	Chapa lisa de acero galvanizado de 0,6 mm de espesor
E44AB25C	Acero S235JRC según UNE-EN 10025-2, para cerchas formadas por pieza simple, en perfiles conformados en frío serie L, U, C, Z y omega, trabajado en taller y galvanizado, colocado en obra con tornillos
E442F13C	Acero S275J0H según UNE-EN 10210-1, para elementos de anclaje formados por pieza simple, en perfiles huecos laminados en caliente serie redondo, cuadrado y rectangular, trabajado en taller y galvanizado, colocado en obra con tornillos
E31524H3	Hormigón para zanjas y pozos de cimentación, HA-30/B/20/Ila+Qa, de consistencia blanda y tamaño máximo del árido 20 mm, vertido con cubilote
E31B3000	Armadura de zanjas y pozos AP500 S de acero en barras corrugadas B500S de límite elástico ≥ 500 N/mm ²
E31D1100	Encofrado con paneles metálicos para zanjas y pozos de cimentación

1.4.2 Programas de cálculo

Los programas utilizados para desarrollar los diversos cálculos del proyecto han sido, ordenados por áreas:

De la parte matemática (movimiento solar)

Para comprobar que el resultado de las ecuaciones implementadas para calcular la posición del Sol ha sido correcto, se ha utilizado la herramienta en línea:

SunEarthTools

(Accesible desde la URL: <http://www.sunearthtools.com/es/>)

De la parte mecánica:

El programa CAD de modelado mecánico en 3D:

SolidWorks.

El cual ha sido utilizado para:

Realizar el diseño preliminar del mecanismo, incluyendo desde las pruebas de movilidad iniciales hasta la determinación de las magnitudes finales que lo compondrán.

Modelar el conjunto de piezas y perfiles utilizadas, su posición final y las restricciones entre ellas.

Llevar a cabo cálculos numéricos para la obtención de momentos de inercia del conjunto.

De la parte estructural:

El programa CYPE Ingenieros, para el cálculo y dimensionamiento de la estructura.

Concretamente, el módulo de generación de pórticos y acciones y el módulo de estructuras tridimensionales.

De la parte de control:

Para la elaboración del programa de control del seguidor se ha utilizado:

Software Arduino 1.6.12

Un programa que implementa una variante de C++ orientada a su implementación en microcontroladores

Además, como programa auxiliar para la detección y corrección de errores, se ha utilizado el programa de cálculo : MATLAB

1.4.3 Plan de gestión de la calidad aplicado durante la redacción del Proyecto

En la parte matemática y de control:

El programa del controlador Arduino ha sido implementado paralelamente con el programa Matlab, para asegurar la validez de las ecuaciones.

Una de las desventajas de los microcotroladores es su precisión relativamente baja, debido a ciertas limitaciones. Como la precisión en algunas variables, las aproximaciones y posibles problemas de desbordamiento de operaciones.

Por este motivo se ha realizado esta doble comprobación, gracias a Matlab, que tiene un poder de cálculo mayor.

Además, una vez comprobado que los resultados en ambos casos coincidían, estos han sido comparados con la información aportada por la página SunEarthTools, para asegurar la validez total de los resultados.

En la parte estructural:

A la hora de llevar a cabo el diseño de la estructura, se han realizado previamente varios cálculos estructurales correspondientes a múltiples posiciones posibles del seguidor.

La definición final de la estructura ha surgido del resultado de esta serie de cálculos discretos, obteniendo cada perfil del estudio correspondiente en que sus solicitaciones eran más desfavorables.

1.4.4 Bibliografía

En lo que respecta a Arduino:

La página oficial del microcontrolador: <https://www.arduino.cc/>
Especialmente la sección de referencias del Lenguaje:
<https://www.arduino.cc/en/Reference/HomePage>
Dónde se puede encontrar la gramática de todas las funciones que trae Arduino por defecto.

También en la página <http://www.prometec.net/>
En la sección Arduino, puesto que cuenta con un amplio tutorial.
Donde cabe destacar los capítulos: 1.1 , 1.2 , 2 (librería Time y relojes RTC , 6.97 , 6.98 , 6.99)

Y la página dedicada a Máquinas de control numérico:

<http://www.productoscnc.es/index.htm>

Haciendo especial mención de las secciones:

Introducción a motores paso a paso:
<http://www.productoscnc.es/hyperpages/unintroamotorestap.htm>

Catálogo de motores paso a paso:
http://www.productoscnc.es/hyperpages/motores_paso_a_paso_pap.htm

1.5 Definiciones y abreviaturas

Explicaciones relativas al movimiento y radiación Solar

Debido a que entre el Sol y la Tierra se encuentra el vacío, la energía que nos llega del Sol lo hace mediante radiación. Ya que este es el único tipo de intercambio de energía que no necesita de un medio físico para llevarse a cabo; al contrario que la conducción y la convección.

El Sol emite energía en todas las direcciones, y una porción de esta llega a la Tierra.

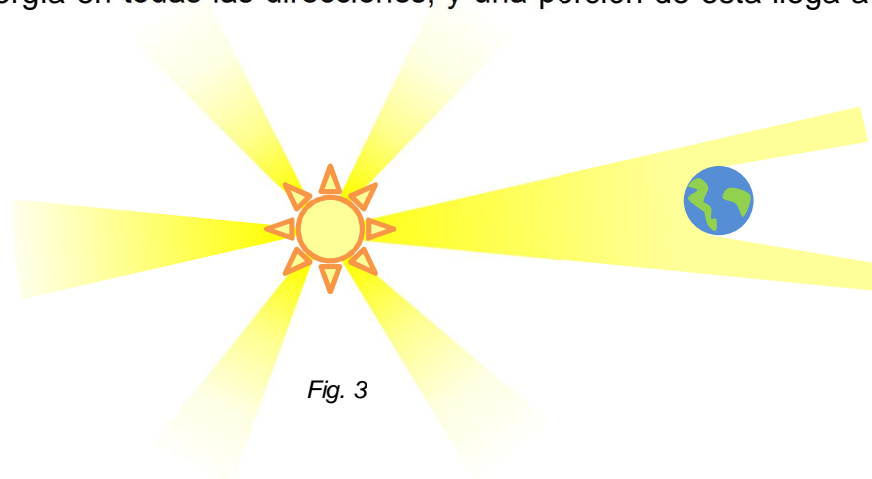


Fig. 3

1. Memoria

A su vez, de la energía que llega a nuestro planeta una parte es frenada por el campo magnético terrestre, otra parte absorbida por la atmósfera y la parte restante llega a la superficie terrestre.

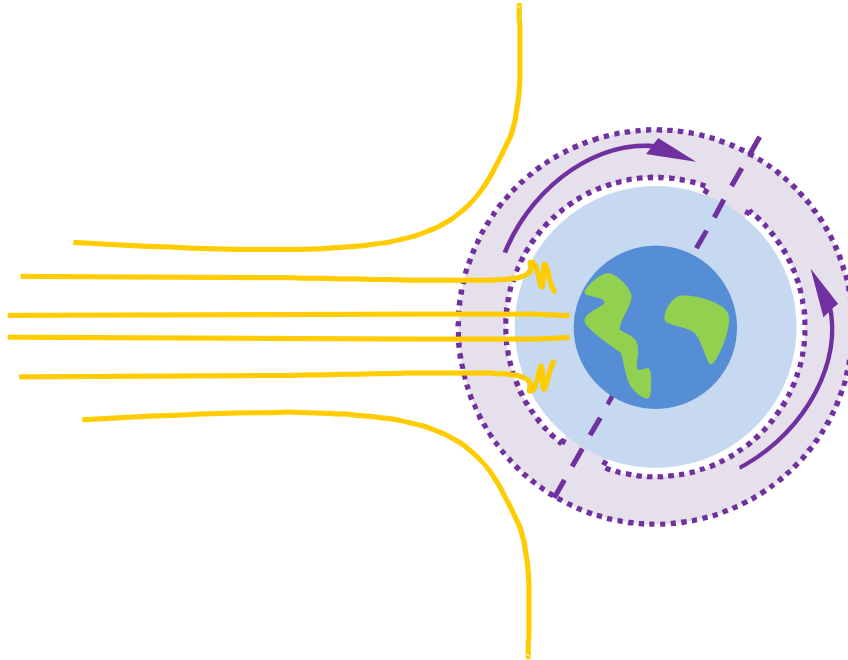


Fig. 4

Además de esto, la cantidad de energía recibida depende de la época del año (ángulo entre los rayos solares y la superficie terrestre) y la situación geográfica (latitud y longitud). Siendo mayor la energía recibida en verano que en invierno, y menor a medida que aumenta la latitud del punto en cuestión.

De aquí extraeremos los parámetros necesarios para nuestro seguidor. Como hemos mencionado anteriormente, llevaremos a cabo el control mediante la implementación de las ecuaciones del movimiento solar. Para las cuales necesitaremos saber: latitud y longitud del punto dónde está instalado el seguidor y día del año.

Cada día, el Sol sale por el Este, a una cierta hora y a un cierto ángulo respecto el Sur geográfico, en este instante la altitud solar es 0°

A medida que pasa el tiempo, se va desplazando hacia el Oeste mientras va aumentando su altitud.

Cuando alcanza el mediodía solar, se encuentra en su altitud máxima. A partir de este momento empieza a descender mientras se sigue desplazando hacia el Oeste.

1. Memoria

A la hora del ocaso, el Sol se pone (altitud 0°) en una posición determinada respecto al Sol geográfico.

Las ecuaciones de movimiento solar nos permiten, a partir del día del año y la posición geográfica del punto de instalación, obtener todos los datos que necesitamos. Los cuales son la hora y ángulo de salida/puesta del Sol para cada día.

Definiciones relativas a la radiación Solar

Como hemos dicho anteriormente, la energía del Sol llega a la tierra en forma de radiación. De esta, solo una parte consigue alcanzar la superficie terrestre y la podemos cuantificar mediante las siguientes magnitudes:

Irradiación (G o H): es la energía radiante que llega a una superficie en un tiempo determinado. Se expresa en unidades de energía, Joule o kWh

Dentro de la Irradiación puede haber de diferentes tipos:

Directa (I): Es aquella que consigue atravesar completamente la atmósfera sin refracciones ni reflexiones intermedias

Difusa (D): Es aquella emitida por el entorno celeste, debida a la reflexión y refracción en la atmósfera. El causante principal de este tipo de radiación son las nubes.

Reflejada (R): Es el nombre que recibe la parte de la radiación directa que es reflejada por una determinada superficie. Nótese que la radiación difusa, al proceder de todas partes no se refleja.

Irradancia (E): se puede definir como la energía que incide por unidad de tiempo y superficie, como la potencia incidente o como la intensidad media de la radiación. Se expresa en W/m^2

La suma de la radiación directa y difusa es lo que llamamos radiación global o irradiación

Irradiación global = Irradiación Directa + Irradiación Indirecta

O simplificado:

$$G = I + D$$

Esta magnitud es la que viene dada por los mapas, medida en el plano horizontal.

Se estima que en buenas condiciones se tienen unos valores de irradiancia de $1000W/m^2$

1. Memoria

En España los valores de irradiación suelen estar entorno unos 1660 kWh/m² anuales, lo que equivale a 4,55 kWh/m²día
Concretamente en Castellón se estiman unos 4,6–5 kWh/m²día

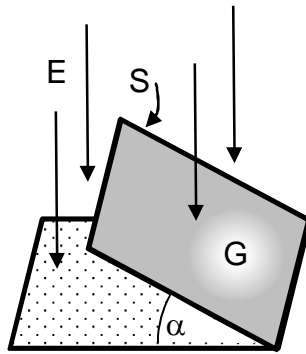


Fig. 5

Si queremos obtener la irradiación sobre una placa de superficie “S”, ambas magnitudes se pueden relacionar mediante la ecuación:

$$G = E \cdot S \cdot \cos \alpha \cdot t$$

Y sus unidades son:

$$(J) = (W/m^2) \cdot (m^2) \cdot (s)$$

Explicación del movimiento solar y definición de sus variables

A la hora de delimitar la posición del Sol en cada momento, necesitaremos conocer las siguientes magnitudes:

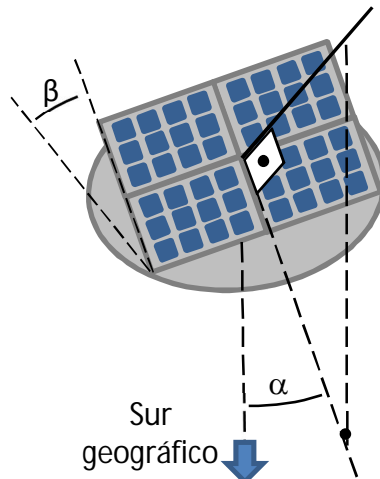


Fig. 6

Orientación (α): Es el ángulo entre la normal a la superficie y la línea que marca el Sur geográfico. También conocido como Azimut.

Inclinación (β): Es el ángulo entre la superficie de interés y la horizontal.

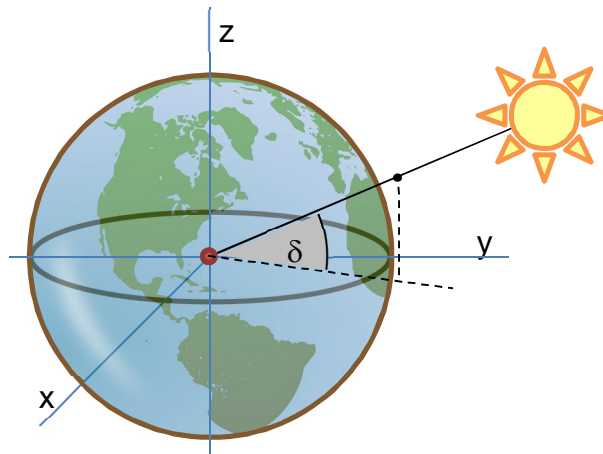


Fig. 7

Declinación (δ): representa el ángulo entre el plano del Ecuador y la recta que une el Sol con el centro de la Tierra. Se puede aproximar por:

$$\delta = 23,45 \cdot \sin \left[360 \cdot \frac{284 + z}{365} \right]$$

Dónde z es el día del año, expresado entre 1 y 365

1. Memoria

Latitud (ϕ): Es el ángulo entre el punto dónde se colocará la placa y el Ecuador

Ángulo horario (h), se calcula como:

$$h = 15 \cdot (12 - TSV)$$

Hora Solar (TSV), se calcula como:

$$TSV = HO - \varepsilon + \frac{ET}{60} + \frac{\lambda_m - \lambda}{15}$$

Dónde:

HO es la **hora oficial** (sobre 24h, siendo las 13:30 las 13,5)

ε es la **corrección horaria** (1h en invierno, 2h en verano)

ET es la ecuación del tiempo (en horas)

λ_m es la **longitud del huso horario del punto de instalación**

λ es la **longitud del punto de instalación** (positivo hacia el oeste)

Ecuación del tiempo (ET, en minutos) es el término corrector que tiene en cuenta la variación de velocidad de rotación de la Tierra.

$$ET = 9,87 \cdot \sin 2B - 7,53 \cos B - 1,5 \sin B$$

siendo $B = \frac{360}{364}(z - 81)$ y z el día del año expresado entre 1 y 365

Nótese que el término de la ecuación del tiempo va dividida entre 60, para obtener el valor total, en horas.

Con todo esto podemos calcular la posición del Sol en cada momento, mediante las ecuaciones:

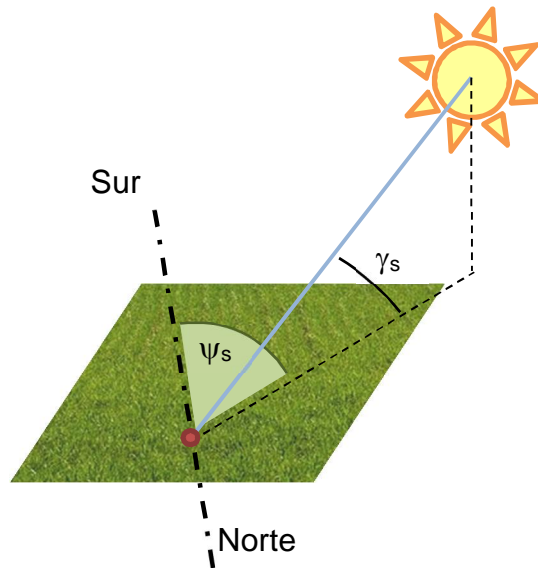


Fig. 8

Altitud Solar (γ_s), representa la elevación del Sol vista desde el punto de instalación. Viene dado por la **Ecuación de la Altitud Solar** y se calcula cómo:

$$\sin(\gamma_s) = \sin(\phi) \cdot \sin(\delta) + \cos(\phi) \cdot \cos(\delta) \cdot \cos(h)$$

Azimut Solar o simplemente Azimut (ψ_s), es el ángulo entre la línea que une el punto de instalación con el Sol y el Sur geográfico. Viene dado por la **Ecuación del Azimut Solar** (o ecuación Azimutal) y se calcula cómo:

$$\cos(\psi_s) = \frac{\cos(\delta) \cdot \sin(h)}{\cos(\gamma_s)}$$

Nota: En el Diccionario de la lengua española, la RAE acepta indistintamente la escritura del término con “c” o con “z”.

Otras definiciones

machacar: relativo a la programación. Actualizar el valor de una variable asignándole un nuevo valor y perdiendo el valor que tenía anteriormente asignado

stepper: motor paso a paso

Cuando se esté hablando en un contexto relativo a la posición del Sol, de las placas solares o del conjunto del seguidor, se podrán utilizar los términos horizontal y vertical para hacer referencia a todo aquel movimiento relacionado con el movimiento acimutal y de elevación, respectivamente.

1.6 Requisitos de diseño

1.6.1. Datos de partida

Por las características del presente proyecto, necesitamos diseñar un mecanismo que, por su constitución y movilidad, permita situar una superficie plana de forma perpendicular a los rayos del Sol en todo momento a lo largo del día.

Para poder empezar a diseñar el mecanismo, es necesario comprender cómo se mueve la Tierra alrededor del Sol. Y por extensión, cómo se aprecia este movimiento visto desde la Tierra.

Por este motivo, la posición del Sol respecto de la Tierra, vista desde un punto en la superficie terrestre depende del día, mes, hora y del punto de observación

Consideremos un punto sobre la superficie de la Tierra, el cuál hemos tomado como origen de un sistema de coordenadas esféricas. (para esclarecer la situación hemos añadido unos ejes cartesianos)

Aparentemente, sobre ese punto puede incidir un rayo de luz desde cualquier dirección.

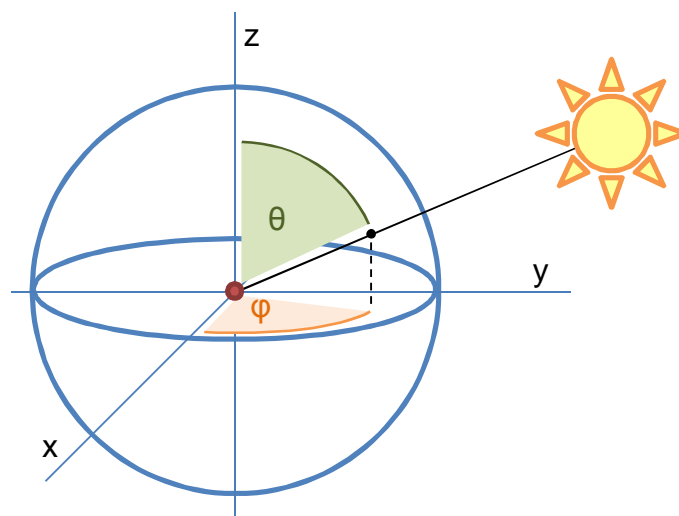


Fig. 9

Debido a la superficie terrestre, la componente θ se encuentra limitada al rango:

$$-90^\circ \leq \theta \leq 90^\circ$$

Es obvio, ya que cuando el Sol está por debajo de la línea del horizonte, no hay ningún punto en la superficie que reciba luz Solar directa.

1. Memoria

En cuanto al ángulo ϕ su rango puede ser como máximo de 360° , claro que el rango máximo únicamente se alcanza en lugares próximos a los polos terrestres y en lugares cercanos al ecuador.

En las zonas cercanas a los polos hay épocas del año en las que es siempre de día y el Sol no llega a ponerse.

En las zonas cercanas al ecuador, debido al ángulo de inclinación del ángulo de rotación de la Tierra, dependiendo de la época del año el Sol sale por el Sudeste y se pone por el Sudoeste o Sale por el Noreste y se pone por el Noroeste

Si en el sistema de referencia esférico coloreamos el volumen que puede ser cubierto por este rango de ángulos, obtenemos un casquete semiesférico. La superficie del cual, contiene todos los puntos a los que los rayos del Sol pueden ser perpendiculares.

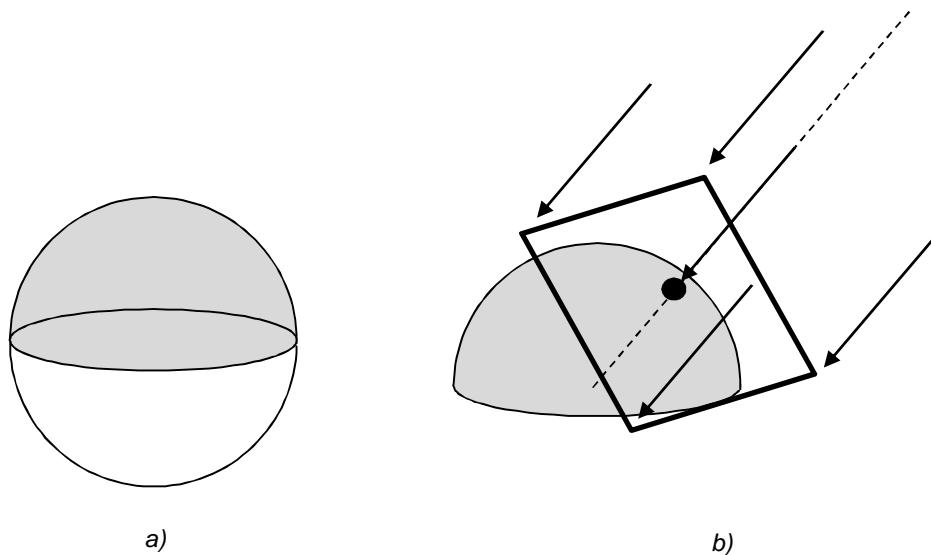


Fig. 10

Esto significa que independientemente de la posición del Sol, siempre habrá un punto contenido en dicha superficie sobre el cual incidan los rayos solares de forma perpendicular

Por tanto, nuestro seguidor debe ser capaz de colocar una placa de forma tangente a cualquiera de los puntos de la superficie de dicho casquete semi esférico.

De este modo tendremos una superficie capaz de ser perpendicular a todas las posibles direcciones de incidencia de los rayos solares

1. Memoria

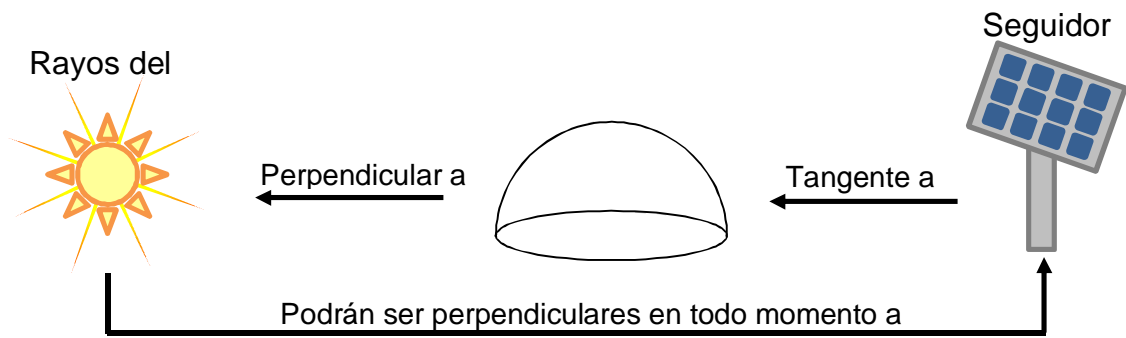


Fig. 11

Como hemos visto, para definir la posición Solar necesitamos dos ángulos, el Azimutal y la Altitud. Lo cual significa que nuestro mecanismo deberá tener dos grados de libertad independientes entre sí.

Ahora que ya sabemos los requisitos que debe tener nuestro seguidor, vamos a buscar un mecanismo que sea capaz de lograrlo.

1.6.2. Emplazamiento

Debido a las características del proyecto y la forma en que ha sido desarrollado se trata de un proyecto muy versátil que podría ser instalado en multitud de lugares.

En nuestro caso particular, vamos a concretar el proceso para los jardines del edificio Espatec II de la UJI.

Cuya dirección es:

Avenida de Vicent Sos Baynat, s/n, 12071 Castellón, Castelló

Su calificación es suelo Urbano de uso Cultural.

1. Memoria

El lugar exacto de emplazamiento será:

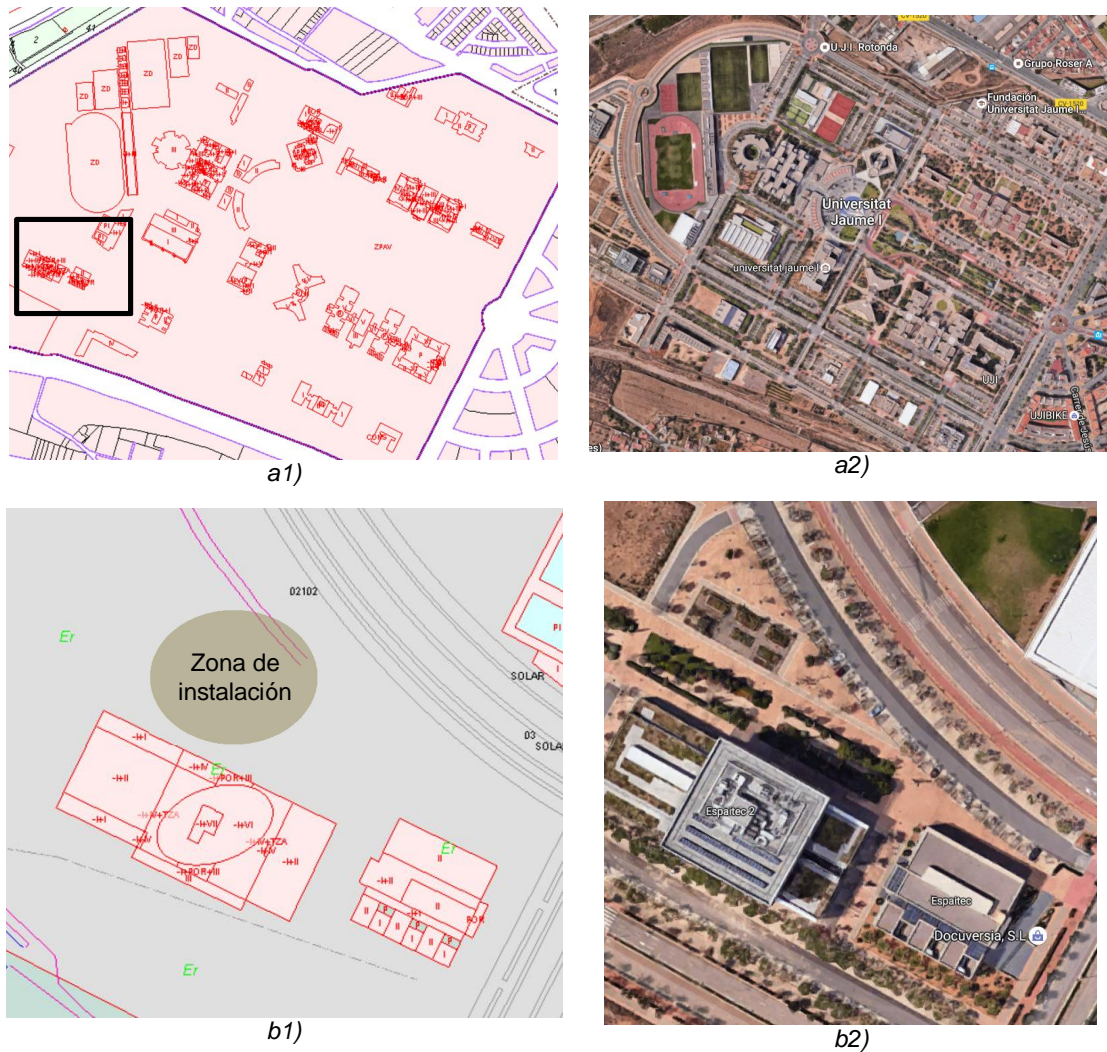


Fig. 12 Emplazamiento

En la figura a1) podemos ver la localización de los jardines de Espeitec II respecto el resto de la universidad. El cuadro marcado se corresponde con la zona ampliada en la figura b1) y en ella podemos ver la zona aproximada dónde será instalado. Ya que el punto exacto de instalación aún no se conoce.

Las imágenes a2) y b2) han sido aportadas a modo ilustrativo, puesto que la delimitación por parcelas es un poco más compleja de visualizar.

1.7 Análisis de soluciones

1.7.1. Seguimiento

Hay múltiples combinaciones de mecanismos con dos grados de libertad que permitirían situar una superficie alineada en todo momento con el Sol.

Dentro de los mecanismos para seguidores solares, hay dos tipos predominantes.

Tipo giroscopio:

Donde el centro de gravedad de las placas permanece inmóvil. Su movimiento es equivalente a una rótula de dos ejes perpendiculares

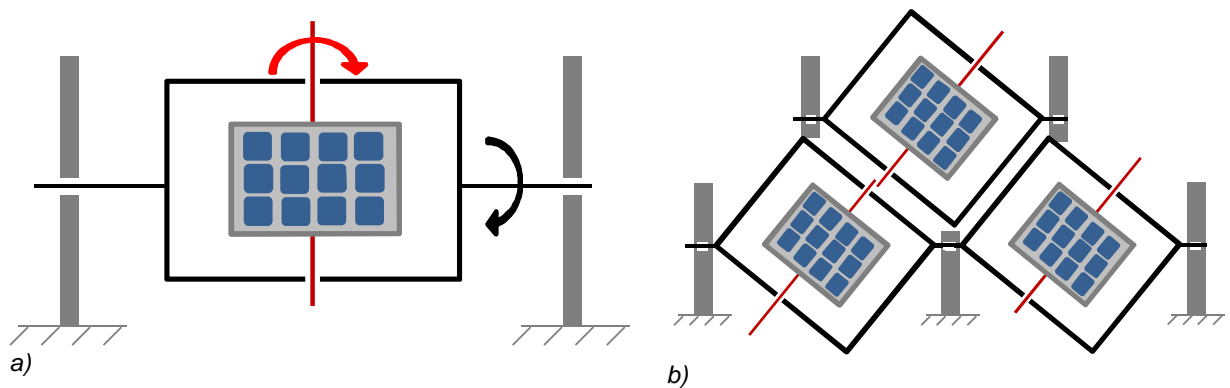


Fig. 13

Este tipo de seguidor resulta más fácil de controlar mediante sensores fotosensibles. Tiene la ventaja de que podemos girar la disposición inicial 45° para reducir considerablemente las sombras mutuas si se instalan varios seguidores en conjunto.

No obstante, tiene el inconveniente de que los actuadores necesitan un aporte de energía constante para mantener la posición de la placa.

El conjunto de placas se monta sobre una estructura de inclinación variable. Y esta, a su vez va apoyada en lo alto de un poste con forma de T.

1. Memoria

Suelen ser algo similar a:

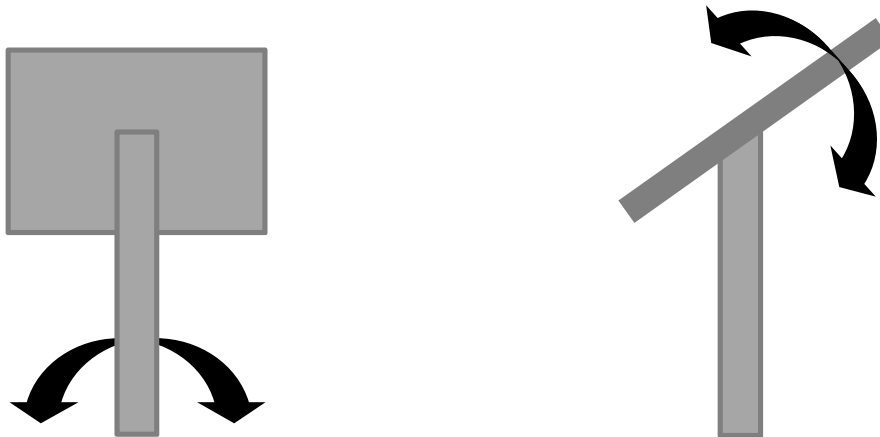


Fig. 14

Dónde las flechas señalan los grados de libertad. Los de este tipo suelen requerir unos anclajes mayores a los anteriores. También requieren un aporte de energía continuo para mantener la posición. No obstante, su control mediante la implementación de las ecuaciones de movimiento solar es bastante más sencillo.

Como hemos podido ver, ambos tipos requieren de un aporte de energía continuo para mantener la posición. Y uno de los objetivos del presente proyecto es conseguir un seguidor que sea capaz de mantener la posición por sí mismo.

Por tanto, debemos pensar en hacer alguna modificación a alguno de estos mecanismos para conseguir implementarles un sistema de autorretención.

El mecanismo en el que hemos pensado es el siguiente:

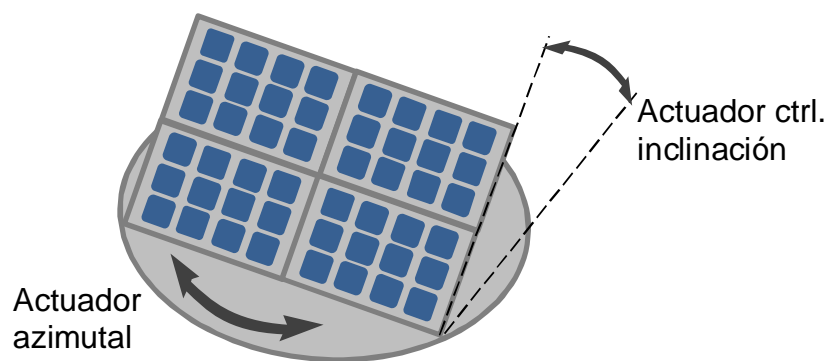


Fig. 15

1. Memoria

Es una evolución del seguidor de tipo heliostato. Solo que en lugar de montar las placas sobre un poste en forma de T, estas serán montadas sobre una superficie horizontal. Y el mecanismo consistirá en un actuador que controlará la rotación de la base horizontal, y otro que controlará la inclinación de las placas.

Este mecanismo es el que hemos elegido para la solución final. El cual podemos expresar mediante el siguiente esquema:

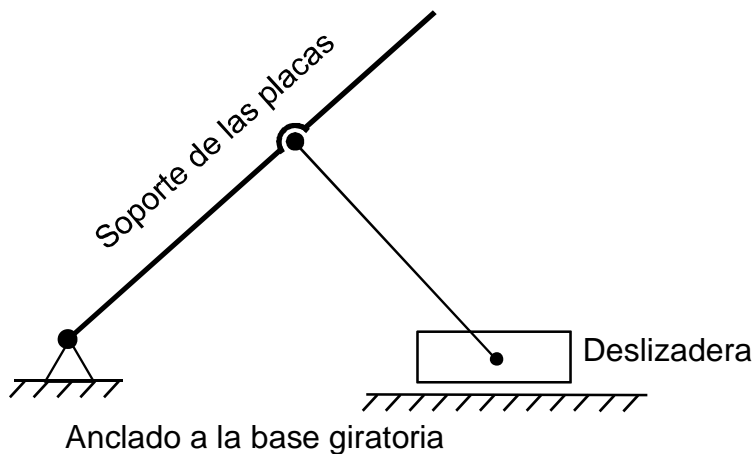


Fig. 16

Vamos a explicar brevemente la movilidad básica del mismo:

Mediante el control de la posición deslizadera, podemos controlar la inclinación de la placa. Con este mecanismo de tres barras controlaremos la inclinación de los paneles.

Finalmente, bastará montar este mecanismo sobre la base giratoria para añadir el movimiento azimutal que necesitamos.

Vamos a demostrar que, efectivamente nuestro mecanismo presenta las características necesarias para llevar a cabo su función.

En cuanto a la inclinación de las placas, vamos a estudiar cómo se comporta el mecanismo. En la Fig. 16 se ha representado para una posición azimutal fija, los dos casos extremos que se pueden dar cuando tengamos la placa totalmente abatida a) o totalmente levantada c)

1. Memoria

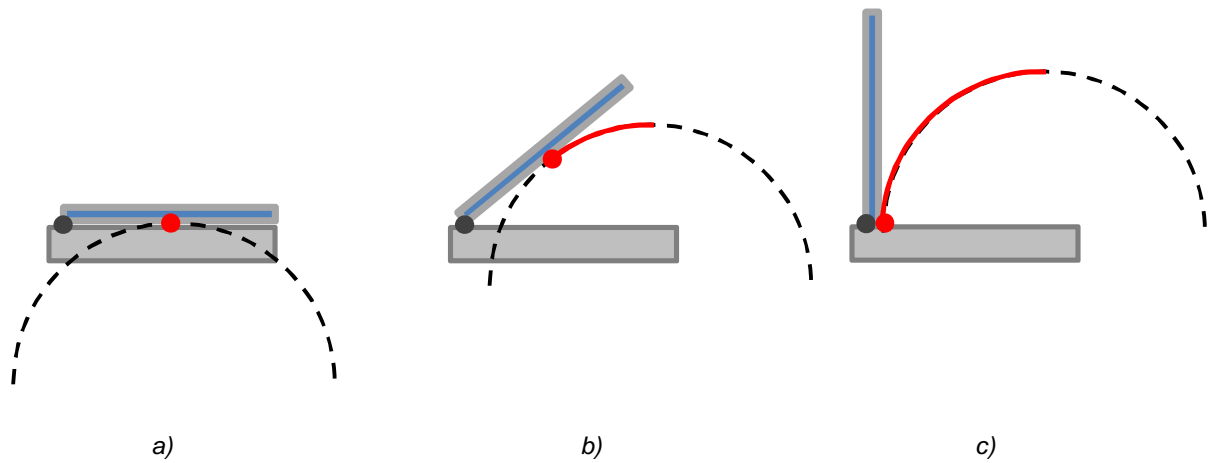


Fig. 17

Para ejemplificarlo mejor, se ha añadido un caso intermedio b).

El punto rojo muestra el punto de la superficie semiesférica a la que es tangente la placa en esa posición, y la línea roja es su trayectoria a medida que la placa se levanta.

Por tanto, para cada una de las posibles posiciones azimutales, el conjunto de puntos a los que la placa puede ser tangente es un cuarto de circunferencia (Fig. 17 a).

Puesto que este mecanismo va montado sobre una base giratoria, cuyo rango de giro es 360° b)

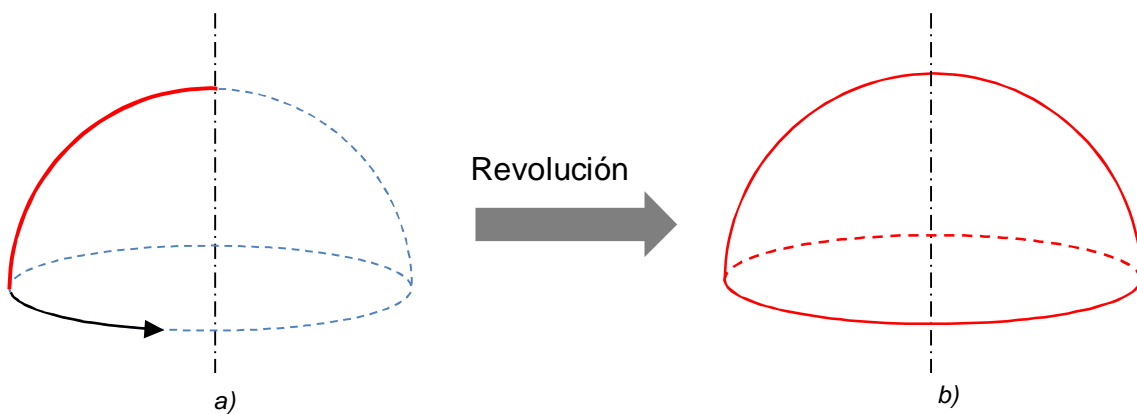


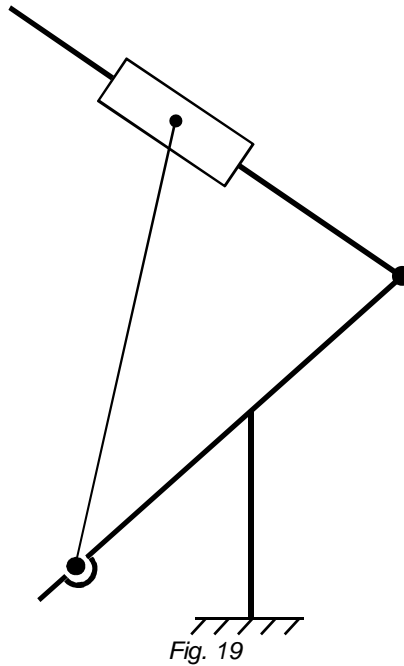
Fig. 18

1. Memoria

Efectivamente nuestro seguidor es capaz de ser tangente a todos los puntos de un casquete semiesférico

Aunque como acabamos de ver, el mecanismo cumple los requisitos de movilidad necesarios, salta a simple vista que el diseño resulta un poco tosco. Ya que se compone de una base de un tamaño considerable, girando con placas colocadas sobre ella.

Por tanto, para estilizar el mecanismo, adoptaremos la siguiente modificación:



En este caso, en lugar de montar la deslizadera sobre la base horizontal, la montaremos tras el soporte de las placas.

1.7.2. Estructura.

En el punto anterior hemos analizado un mecanismo de forma bidimensional, no obstante, nuestro seguidor será una estructura tridimensional y deberemos adaptar el mecanismo obtenido.

Lo que haremos será montar dos mecanismos iguales en sendos lados de una barra transversal que irá unida al mástil:

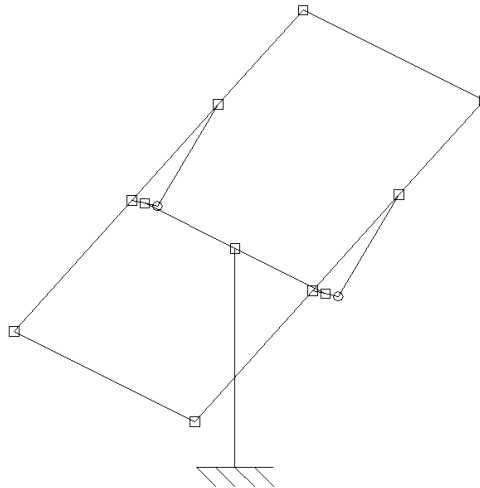


Fig. 20

Como la parte superior girará, hemos decidido construir el mástil con un perfil circular, de este modo tendremos la certeza de que se comportará del mismo modo sea cual sea la posición del seguidor.

La barra transversal la construiremos a partir de un perfil tubular con dos planchas en los extremos a modo de aletas, que se corresponden con la barra fija inclinada del mecanismo en la figura 19.

Sobre esas aletas montaremos los dos mecanismos descritos, que constarán de una barra de perfil tubular cuadrado, cuyo extremo se deslizará de forma paralela al armazón de sujeción de las placas.

Los perfiles seleccionados para los laterales del armazón serán del tipo IPE, ya que principalmente estarán sometidos a flexión.

Además se le añadirán dos montantes de perfil cuadrado para hacer más rígido el conjunto.

Finalmente, las placas irán unidas al armazón mediante cuatro correas con perfil en forma de C. Aunque un perfil tubular tendría mejores características, hemos elegido este tipo de correas porque permiten atornillar las placas más fácilmente.

Todas las uniones fijas irán soldadas salvo que se especifique lo contrario.

1.8 Resultados finales

1.8.1. Mecanismo

Para hacer más fácil la programación, los movimientos elegidos serán dos rotaciones, una horizontal y la otra vertical. La unión de estos dos movimientos, independientes entre sí, nos permitirá alinear los módulos solares en cualquier posición.

En el caso del movimiento de rotación horizontal usaremos directamente un motor giratorio con reductora que estará situado en una junta giratoria en la parte alta del mástil

En cambio, para realizar la elevación de la placa utilizaremos un motor giratorio conectado a un tornillo sinfín, lo cual nos permitirá transformar el movimiento angular del motor en movimiento lineal.

La placa la montaremos sobre la guía de un mecanismo de tres barras con deslizadera. Con el actuador lineal controlaremos la posición relativa entre la guía y la deslizadera, lo cual influirá en la inclinación del conjunto.

1.8.2. Estructura

Para llevar a cabo el dimensionamiento se ha utilizado el programa CYPE. Los elementos principales de la estructura final son:

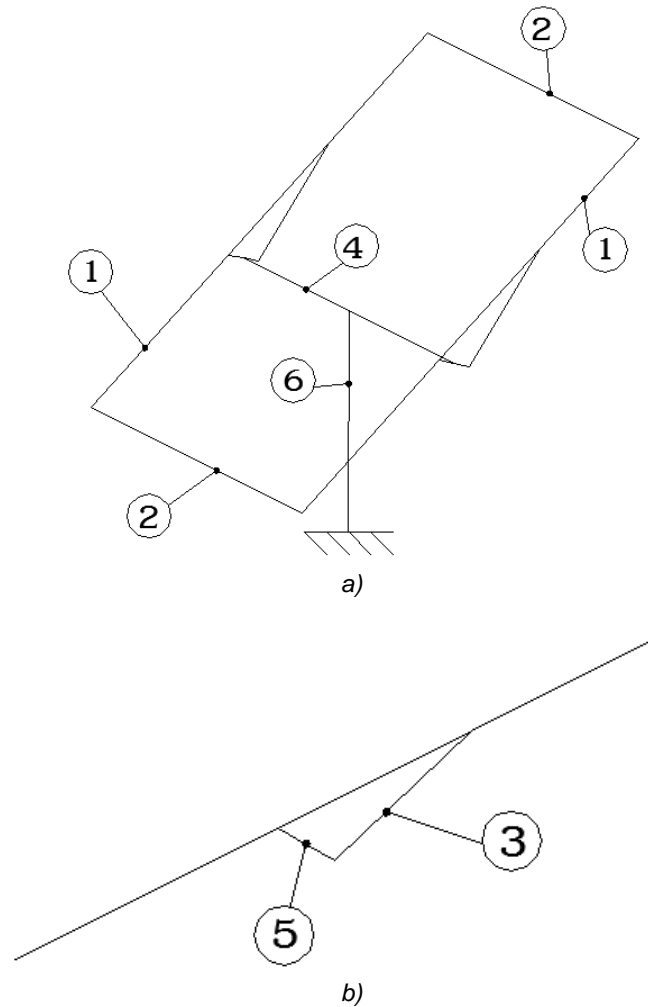


Fig. 21

Marca	Barra	Unidades	Comentarios
1	Larguero	2	Son las dos barras laterales
2	Montante	2	Junto los anteriores forman el marco del armazón
3	Barra móvil	2	Será la utilizada para variar la inclinación de las placas
4	Barra central	1	Sobre sus extremos se montan los mecanismos de inclinación de las placas
5	Aleta	2	Van unidas a los extremos de 4
6	Mástil	1	Pilar central, soporte del conjunto

1. Memoria

Tras realizar los cálculos, los resultados obtenidos han sido:

Barra	Tipo perfil	Material	Longitud (m)	Volumen (cm ³)	Peso (kg)
Larguero izquierdo	IPE 100	Acero S275	3,2	3302,4	25,92
Larguero derecho	IPE 100		3,2	3302,4	25,92
Montante superior	TUBO 20 20 2		2	266	2,09
Montante inferior	TUBO 20 20 2		2	266	2,09
Barra móvil izquierda	TUBO 30 20 3		0,93	223,2	1,75
Barra móvil derecha	TUBO 30 20 3		0,93	223,2	1,75
Barra central	TUBO 120 120 5		2	4468	35,07
Aleta izquierda	PL 120 8		0,33	316,8	2,49
Aleta derecha	PL 120 8		0,33	316,8	2,49
Mástil tubular	CHS 200 4.8		1,6	5161,6	40,52
Correa 1	UL 65 42		2	1640	12,81
Correa 2	UL 65 42		2	1640	12,81
Correa 3	UL 65 42		2	1640	12,81
Correa 4	UL 65 42		2	1640	12,81

1.8.3. Placa anclaje

Placas de anclaje				
Material	Elementos	Cantidad	Dimensiones (mm)	Peso (kg)
S275	Placa base	1	350x350x14	13,46
	Total			13,46
B 500 S, Ys = 1,15 (corrugado)	Pernos de anclaje	4	Ø 16 - L = 400	2,53
	Total			2,53

1.8.4. Cimentación.

Referencia: N14		B 500 S, CN	Total
Nombre de armado		Ø12	
Parrilla inferior - Armado X	Longitud (m)	9x2,15	19,35
	Peso (kg)	9x1,91	17,18
Parrilla inferior - Armado Y	Longitud (m)	9x2,15	19,35
	Peso (kg)	9x1,91	17,18
Parrilla superior - Armado X	Longitud (m)	9x2,15	19,35
	Peso (kg)	9x1,91	17,18
Parrilla superior - Armado Y	Longitud (m)	9x2,15	19,35
	Peso (kg)	9x1,91	17,18
Totales	Longitud (m)	77,40	
	Peso (kg)	68,72	68,72
Total con mermas (10.00%)	Longitud (m)	85,14	
	Peso (kg)	75,59	75,59

1.8.5. Selección de actuadores y transmisión.

Movimiento de elevación:

El actuador seleccionado para llevar a cabo la elevación de las placas es un motor paso a paso de 3,1 Nm, Nema23 M23-3.1-1D10-D2-A

Para sistema de transmisión por tornillo de potencia hemos elegido una tuerca de bronce de tipo EFM de TECNOPOWER, concretamente el modelo EFM Tr 40x7 y su correspondiente husillo, el RPTS Tr 40x7 que en nuestra aplicación deberá medir 600mm de longitud.

Solución alternativa:

Se podrían sustituir los dos motores de elevación por uno solo, añadiendo una transmisión que lo permita. Esto podría hacerse mediante un eje y engranajes cónicos o por transmisión con cadena. Habría que hacer una valoración de eficiencia y peso, tanto del nuevo motor como de la transmisión seleccionada.

Movimiento de rotación

El actuador seleccionado es un motor Nema34 de 4.2 Nm de par motor, concretamente el modelo M34-4.2-1SB12-E2-A con una reductora de 1/20

Para la transmisión hemos elegido un tornillo sinfín con engranaje de diámetro interior 250mm, lo cual permitirá su montaje en el propio tubo. Concretamente el modelo AGF6-60R1 de la marca KHK y su piñón correspondiente, el KWG6R1 con una reductora de 1/4

1.8.9. Enumeración de cuestiones pendientes de resolver

- Dimensionado de las uniones móviles del mecanismo.
- Selección de las reductoras a instalar entre el motor y la transmisión
- Comprobación minuciosa de la transmisión
- Selección de los acoples rígidos motor-reductora y reductora-transmisión
- Diseño de un sistema de alimentación normal y otro para evitar perder el valor de las variables en caso de fallo.
- Incorporar una forma de situar manualmente el seguidor en una posición dada, para facilitar la inicialización.
- Proponer alternativas de materiales y configuraciones para aligerar la estructura
- Adaptar el programa para su correcto funcionamiento en zonas ecuatoriales y cercanas a los polos

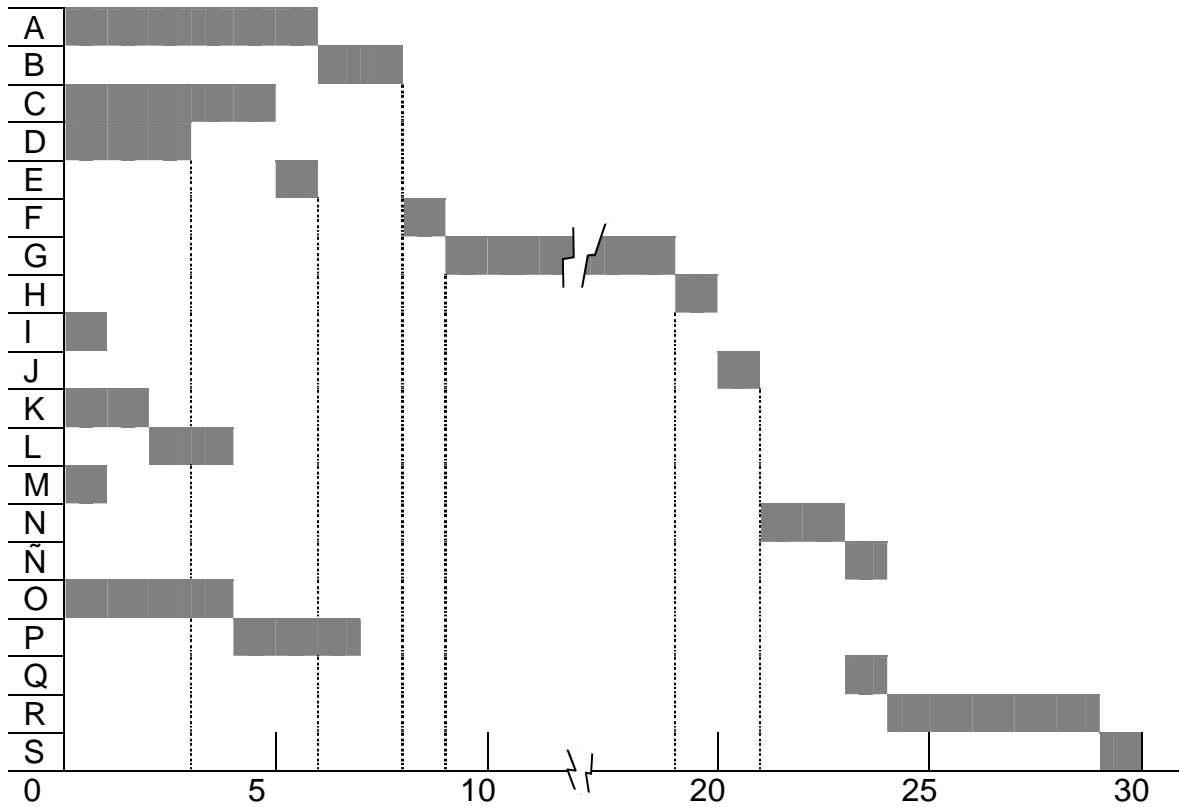
1.9 Planificación

La secuencia de materialización del proyecto será:

Abreviatura	Proceso	Duración	Precedentes
A	Excavación zanja	6	-
B	Preparar armazón	2	A
C	Fabricación del mástil	5	-
D	Fabricación placa de anclaje	3	-
E	Unión del mástil y la placa	1	C, D
F	Vertido del cemento	1	B
G	Fragua	10	F
H	Instalación del mástil	1	G, E
I	Fabricación barra transversal	1	-
J	Unión de la barra transversal al mástil	1	I, H
K	Unión de los largueros y montantes (marco)	2	-
L	Unión del marco y las correas (bastidor)	2	K
M	Preparación barra móvil	1	-
N	Montaje de la estructura	2	J, L, M
Ñ	Instalación de las placas	1	N
O	Realizar los circuitos impresos	4	-
P	Conexión y montaje en la caja de empalmes	3	O
Q	Montaje placa trasera	1	N
R	Instalación de los actuadores y sensores	5	Q
S	Conexión final	2	P, R

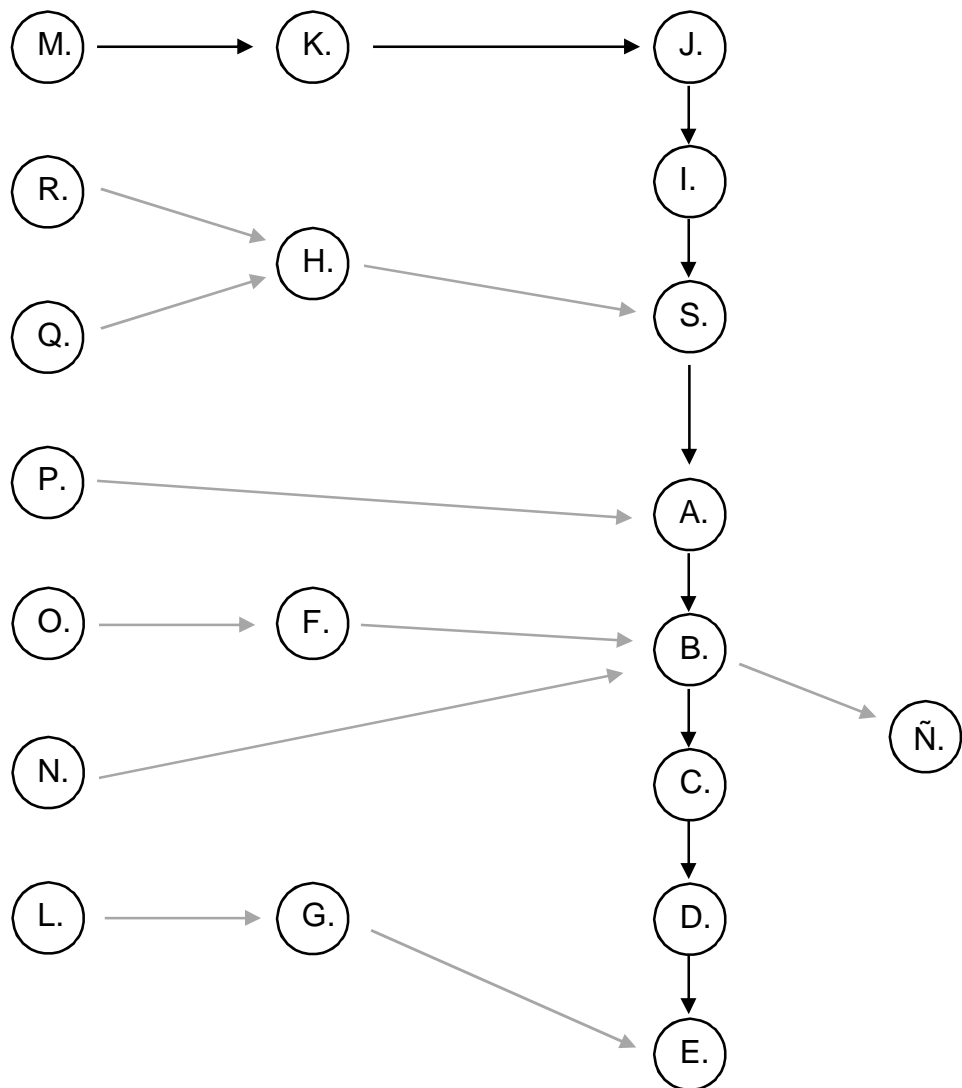
**Los tiempos son orientativos*

1.9.1. Diagrama de Gantt



La operación G tarda 10h, pues como podemos ver va desde las 9 a las 19 horas. Se muestra con el eje partido para compactar el diagrama

1.9.2. Diagrama de redes y camino crítico



Todas las operaciones que no necesitan de ninguna otra para poder ser realizadas aparecen la columna izquierda.

El camino crítico es el delimitado por las flechas más oscuras.

1. Memoria

1.10 Orden de prioridad entre los documentos

En caso de discrepancia en el contenido del presente proyecto, el orden de prioridad entre documentos será el siguiente:

- 1º. Anejos a la Memoria y Memoria
- 2º Planos
- 3º Pliego de condiciones
- 4º Presupuesto

1. Memoria

2. Anejos a la memoria

2.1. Estrategia de seguimiento y resolución

Los pasos (simplificados) que llevará a cabo el autómata para realizar el seguimiento del Sol serán los siguientes:

- 1- Cálculo de la hora y azimut de la salida y puesta del Sol
- 2- Colocación en la posición correspondiente a la salida del Sol
- 3- A partir de la salida del Sol, re-calcular su próxima posición.
- 4- Una vez llegada la hora de puesta del Sol, detención de la rutina 3 y vuelta al punto 1.

Proceso de cálculo:

Sabemos que a la hora del amanecer y anochecer la altitud solar es, por definición, igual a 0. Por tanto, a partir de la ecuación de la altitud, obtenemos:

$$0 = \sin(\phi) \cdot \sin(\delta) + \cos(\phi) \cdot \cos(\delta) \cdot \cos(h)$$

En la que ϕ (latitud) es constante todo el tiempo, y δ (declinación) depende directamente del día del año.

Por tanto podemos obtener el ángulo horario (h) como:

$$h = \cos^{-1} \left(\frac{-\sin(\phi) \cdot \sin(\delta)}{\cos(\phi) \cdot \cos(\delta)} \right)$$

De aquí obtendremos dos valores para h, uno comprendido entre 0° - 180° y el otro comprendido entre 180° - 360° cada uno de ellos estará asociado con la salida o puesta del Sol.

Una vez obtenido h, lo usaremos para calcular las dos horas solares:

$$TSV = 12 - \frac{h}{15}$$

Finalmente con las horas solares obtenemos la hora Oficial de salida y puesta del Sol:

$$HO = TSV + \varepsilon - ET + \frac{\lambda - \lambda_m}{15}$$

Cabe recordar que TSV lo acabamos de calcular, ε y ET dependen del día del año y ambos λ dependen de la longitud geográfica del punto de instalación.

Podemos obtener las horas de Sol (T_{Sol}) de ese día mediante:

2. Anejos a la memoria

$$T_{Sol} = HO_{puesta} - HO_{salida}$$

A continuación, necesitamos calcular el ángulo azimutal de salida y puesta del Sol:

$$\cos(\psi_s) = \frac{\cos(\delta) \cdot \sin(h)}{\cos(\gamma_s)} = \frac{\cos(\delta) \cdot \sin(h)}{\cos(0)} = \cos(\delta) \cdot \sin(h)$$

Lo obtendremos a partir de magnitudes que previamente hemos calculado.

Puesto que tenemos dos valores de ángulo solar, obtendremos la posición de salida y puesta del Sol.

Y, a su vez deberemos ir calculando la altitud solar para cada instante con la ecuación:

$$\text{sen}(\gamma_s) = \sin(\phi) \cdot \sin(\delta) + \cos(\phi) \cdot \cos(\delta) \cdot \cos(h)$$

Este proceso aparece completo y con su implementación en el apartado:

2.7. Descripción del programa

2.2. Descripción del mecanismo

Este tipo de seguidor resulta bastante más fácil de programar en general, puesto que la programación basada en sensores fotosensibles es muy similar al caso anterior. En ese sentido, aquí la gran ventaja reside en implementar las ecuaciones de movimiento Solar. Ya que con esta configuración, la rotación horizontal se corresponde con el azimut solar, y la inclinación de las placas va asociada a la altitud solar.

Además, el actuador encargado de llevar a cabo la rotación horizontal (azimutal) no necesita un aporte de energía para mantener su posición.

Este segundo tipo es el mas adecuado para nuestro caso, puesto que el control lo llevaremos a cabo mediante las ecuaciones solares. Además, para reducir el consumo de energía buscaremos algún dispositivo de autorretención para evitar tener que estar proporcionando energía al actuador encargado de la inclinación de las placas

En cuanto al estudio cinético, consideraremos que sus efectos son despreciables, por la baja velocidad a la que se moverá el seguidor.

Vamos a calcularlo para las coordenadas de los jardines del edificio Espatec II de la UJI: 39.994266, -0.073785

Para ello necesitaremos la siguiente información:

Día: día y mes de los datos

As: Ángulo azimutal en el instante de la salida del Sol, en grados

Ap: Ángulo azimutal en el instante de la puesta del Sol, en grados
(Tomando el Norte como 0° y el Este como 90°)

Al: Altitud solar máxima (°)

Hs: Hora de la salida del Sol, en formato 24h (hh:mm:ss)

Hp: Hora de la puesta del Sol, en formato 24h (hh:mm:ss)

Tr': Tiempo que está el Sol fuera, en segundos

Día	As (°)	Ap(°)	Ar(°)	Al(°)	Hs(24h)	Hp(24h)	Tr' (s)
21-jun	57,89	302,1	244,21	73,44	5:31:42	20:32:42	54060
21-dic	120,46	239,53	119,07	26,56	8:18:45	17:38:29	33584
21-mar/sep	88,75	271,51	182,76	50,52	7:01:39	19:13:43	43924

A partir de estos datos, podemos calcular el ángulo azimutal recorrido en un día por el Sol (Ar), en grados cómo:

$$Ar = Ap - As$$

También podemos obtener el número de horas de Sol de ese día como:

$$Tr = Hp - Hs$$

2. Anejos a la memoria

Obtendremos un valor de Tr en formato hh:mm:ss, para obtener Tr' pasamos este valor a segundos.

A continuación, calculamos la velocidad media azimutal que deberá llevar ese día el seguidor (ω_{Az}), en grados por segundo como:

$$\omega_{Az} = \frac{Ar}{Tr'}$$

Finalmente calcularemos la velocidad media a la que variará la inclinación de la placa (ω_{Al}) en grados por segundo.

Nótese que el ángulo de la placa solar y la altitud solar son ángulos complementarios:

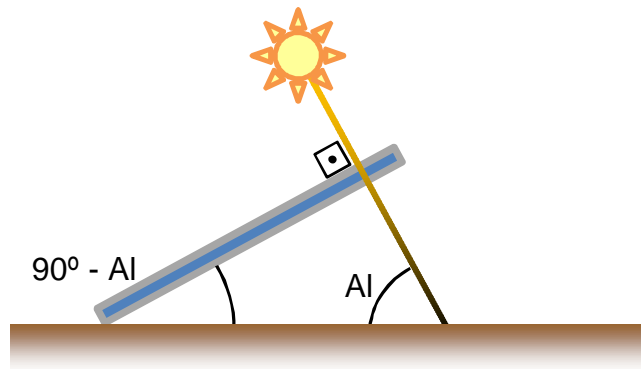


Fig. 22

También hay que tener en cuenta que en un día, la inclinación de la placa asciende hasta el punto máximo y luego desciende otra vez. Por tanto la distancia angular total recorrida es dos veces el ángulo máximo que alcanza la placa. La calculamos como grados por segundo como:

$$\omega_{Al} = \frac{2 * (90 - Al)}{Tr'}$$

Finalmente, pasamos las velocidades a grados por hora para tener la velocidad con unas magnitudes más manejables

Día	ω_{Az}	ω_{Az}	ω_{Al}	ω_{Al}
21-jun	4,52E-03	16,26259711	6,13E-04	2,20554939
21-dic	3,55E-03	12,76357789	3,78E-03	13,60076227
21-mar/sep	4,16E-03	14,97896366	1,80E-03	6,471541754

Nota: Esto son velocidades medias, y por tanto es una aproximación. Para la velocidad azimutal la aproximación es bastante buena, no obstante para la altitud no tanto, debido a que la trayectoria real es una senoide modificada. En el controlador se implementarán las ecuaciones exactas, los cálculos de este apartado son orientativos.

2. Anejos a la memoria

Como podemos observar, las velocidades son muy pequeñas. Tanto, que lo que haremos será mover el mecanismo en pequeños intervalos de posición cada determinado tiempo. Así nos ahorraremos colocar un gran mecanismo reductor de velocidad (con el ahorro de dinero, peso y volumen que eso conlleva) y además al no tener que estar aportando energía de forma continuada también ahorramos energía.

Además, al realizar ambos movimientos en intervalos podemos programarlo para que no se realicen de modo simultáneo, sino consecutivo. Es decir, cada determinado tiempo, en lugar de mover todo el seguidor, hacemos que trabaje primero un actuador y justo después el otro. Con esto conseguimos evitar aceleraciones compuestas y velocidades de arrastre.

Todo esto, junto con las bajas velocidades nos permite despreciar los efectos cinéticos en el mecanismo.

En su lugar, realizaremos un análisis dinámico para determinar las fuerzas de arranque de los motores

2.3. Descripción de actuadores

En primer lugar debemos seleccionar el tipo de actuadores que utilizaremos para controlar cada grado de libertad.

En ambos movimientos necesitamos un sistema que nos permita realizar un pequeño desplazamiento a poca velocidad y con un elevado par. Por tanto nos centraremos la búsqueda en mecanismos que permitan grandes reducciones de velocidad

Para controlar el giro azimutal utilizaremos un motor con tornillo sinfín unido a una rueda dentada, cuyo eje irá unido a la base giratoria.

Para evitar tener que montar el tornillo sinfín, la rueda dentada, y evitar problemas en la transmisión, nos centraremos en modelos comerciales que proporcionan este sistema remontado. Seleccionaremos el modelo que necesitemos dentro de la familia de motores con reductora de elevado par. Como puede ser:

Además, entre la base giratoria y la base fija colocaremos rodamientos para minimizar el par de arranque.

En cuanto el movimiento de inclinación de los paneles, utilizaremos un sistema de varilla roscada con tuerca roscada unida a una deslizadera. de este modo convertiremos el movimiento giratorio en una translación muy lenta.

2. Anejos a la memoria

Y mediante un sistema de dos barras con deslizadera transformaremos este movimiento lineal en inclinación. El diagrama de barras sería:

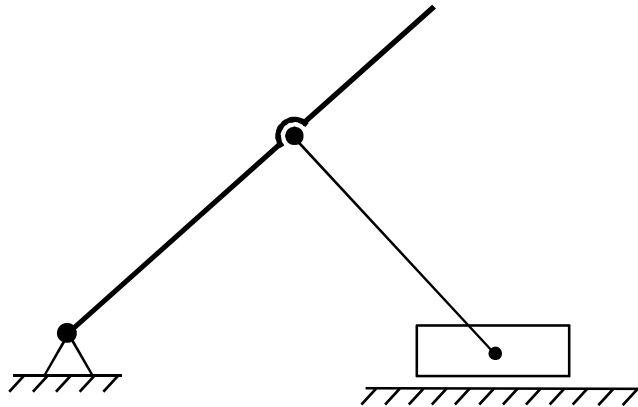


Fig. 23

Además, añadiremos unas barras a modo de guía, que a su vez ayudarán a soportar el peso de la estructura. Así el tornillo soportará el menor peso posible y evitaremos que se deforme.

En cambio, el mecanismo implementado en el seguidor sería:

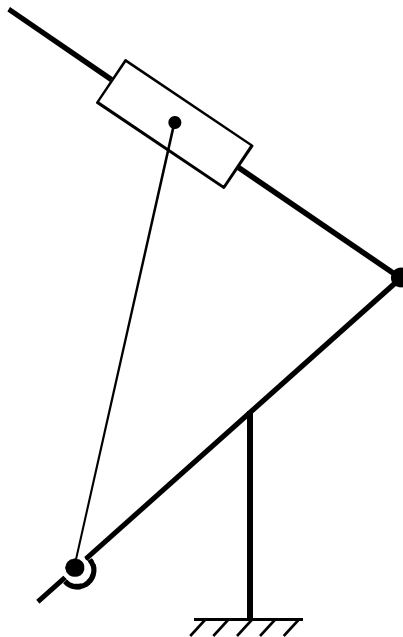


Fig. 24

2.4. Análisis del mecanismo de elevación

A continuación, vamos a calcular la inclinación de la placa en función de todas las magnitudes físicas necesarias del mecanismo, ya que lo necesitaremos para implementar el control del mecanismo

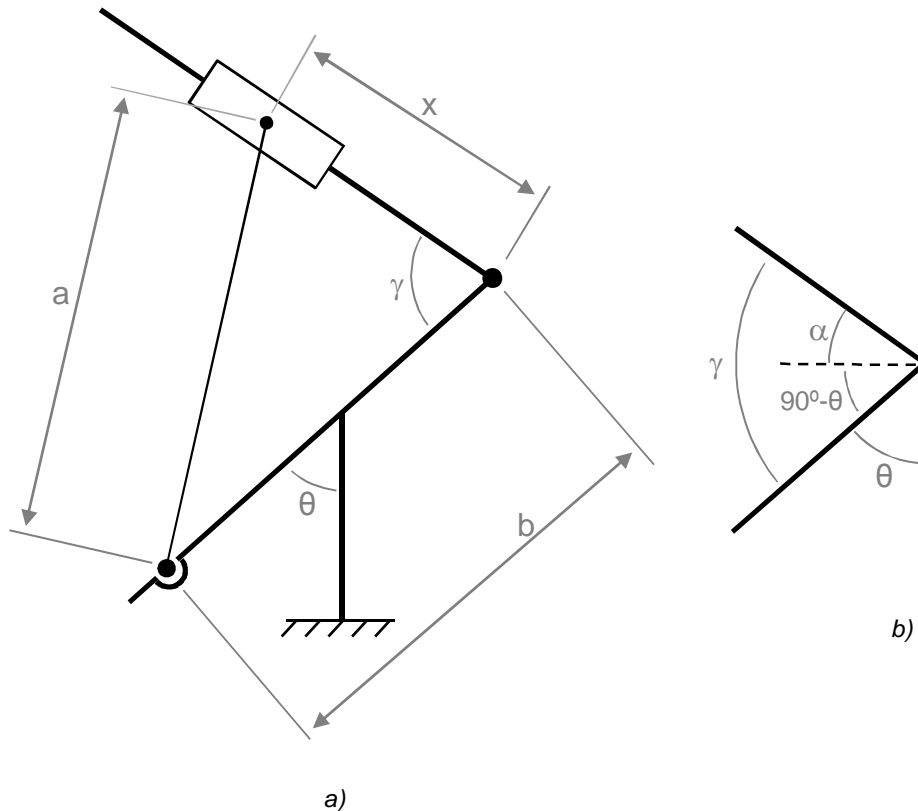


Fig. 25

Las magnitudes:

a , b y θ son características físicas del seguidor, que una vez hayan sido elegidas no podrán ser cambiadas

x y γ (y α a su vez) son las dos variables de posición del mecanismo.

A la hora de controlar el seguidor, para cada instante de tiempo la placa deberá estar a una inclinación (α) para ello deberemos calcular la elongación del tornillo sinfín. Por tanto necesitamos una ecuación que relacione ambas magnitudes: $x = f(\alpha)$

Si aplicamos el teorema del coseno en la figura X:

$$a^2 = b^2 + x^2 - 2bx \cos \gamma$$

2. Anejos a la memoria

Reorganizando en función de x , obtenemos:

$$x^2 - 2bcos\gamma \cdot x + b^2 - a^2 = 0$$

y despejando x :

$$x = \frac{2bcos\gamma \pm \sqrt{(2bcos\gamma)^2 - 4(b^2 - a^2)}}{2}$$

A partir de la figura Y podemos obtener γ en función de α :

$$\gamma = \alpha + 90 - \theta$$

Y sabiendo que la inclinación de las placas y la altitud solar son ángulos complementarios:

$$\alpha = 90 - altitud_{solar}$$

Obtenemos:

$$\gamma = 180 - altitud_{solar} - \theta$$

Con ambas ecuaciones podemos obtener la elongación necesaria del actuador lineal para que la placa se encuentre en la posición solicitada en cada instante.

El valor de x viene dado por una ecuación de 2º grado, por tanto para cada valor de α pueden darse varios casos

2.4.1. Estudio de limitaciones físicas en función de los parámetros del mecanismo:

Para llevarlo a cabo estudiaremos el discriminante (Δ) de la función

$$\Delta = (2b \cos \gamma)^2 - 4(b^2 - a^2)$$

Si $\Delta > 0$

Habrán dos soluciones reales, x_1 y x_2 . Su significado físico son dos valores de x para los que se consigue la inclinación requerida.

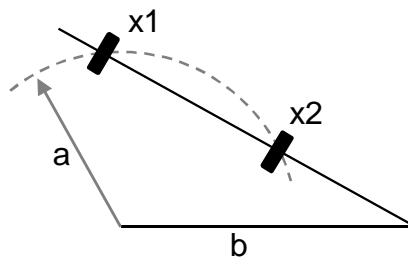


Fig. 26

Si $\Delta = 0$ tenemos una solución doble:

$$(2b \cos \gamma)^2 - 4(b^2 - a^2) = 0$$

$$(2b \cos \gamma)^2 = 4(b^2 - a^2)$$

$$4b^2 \cos^2 \gamma = 4b^2 - 4a^2$$

Dejamos la b y la a una a cada lado de la ecuación y simplificamos los cuatros:

$$b^2 (\cos^2 \gamma - 1) = -a^2$$

Analizamos los valores de a y b para los valores del coseno : $\{0, 0.5, 1\}$

Para $\cos \gamma = 0$:

$$b^2 (0 - 1) = -a^2$$

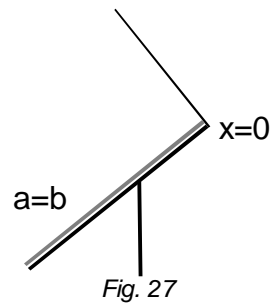
$$-b^2 = -a^2$$

2. Anejos a la memoria

Su significado físico:

$$a = b$$

$$\gamma = 90^\circ$$



Si en nuestro seguidor a y b tienen la misma longitud, y queremos poner la placa en posición perpendicular a b y a , habrá una única solución: $x=0$. En este estado a y b son colineales.

Para $\cos\gamma = 0.5$

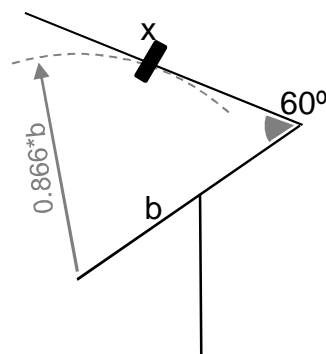
$$b^2 (\cos^2\gamma - 1) = -a^2$$

$$b^2 (1 - 0.5^2) = a^2$$

Haciendo la raíz de ambos miembros, obtenemos la relación entre ambas dimensiones:

$$b \frac{\sqrt{3}}{2} = a$$

Y para un ángulo de $\gamma = \arccos(0.5) = 60^\circ$



2. Anejos a la memoria

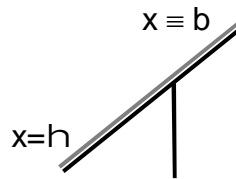
Para $\cos\gamma = 1$

$$b^2 (\cos^2\gamma - 1) = -a^2$$

$$b^2(1 - 1) = -a^2$$

$$0 = -a^2$$

El valor de a debería ser 0 para que $\gamma = \arccos(1) = 0^\circ$



Puesto que la barra a es necesario que esté, su longitud no puede ser 0. El significado físico de esto es que para hacer la placa y la barra b coincidentes, la deslizadera debería alejarse una distancia extremadamente elevada.

La conclusión de este apartado es que utilizando la ecuación:

$$b^2 (\cos^2\gamma - 1) = -a^2$$

Podemos obtener el ángulo para el cual el valor de x tiene un único valor, γ_{limite}

Este valor debe ser comprobado una vez los parámetros a y b sean elegidos puesto que el programa de control estará esperando dos valores de x para cada ángulo α . Por tanto, para ángulos cercanos a γ_{limite} se deberá buscar un método de cálculo alternativo.

Si $\Delta < 0$

Si el discriminante es negativo, no habrá solución. Por tanto vamos a comprobar el rango de valores en los que sí habrá solución ($\Delta > 0$)

$$(2b\cos\gamma)^2 - 4(b^2 - a^2) > 0$$

$$(2b\cos\gamma)^2 > 4(b^2 - a^2)$$

$$4b^2\cos^2\gamma > 4b^2 - 4a^2$$

$$\cos^2\gamma > \frac{4b^2 - 4a^2}{4b^2}$$

$$\cos^2\gamma > \frac{b^2 - a^2}{b^2}$$

2. Anejos a la memoria

Teniendo en cuenta que la función coseno puede tomar valores entre 0 y 1, tomaremos el caso mas desfavorable, cuando el coseno toma su valor mínimo.

$$0 > \frac{b^2 - a^2}{b^2}$$

$$0 > b^2 - a^2$$

$$a > b$$

Veamos su significado físico:

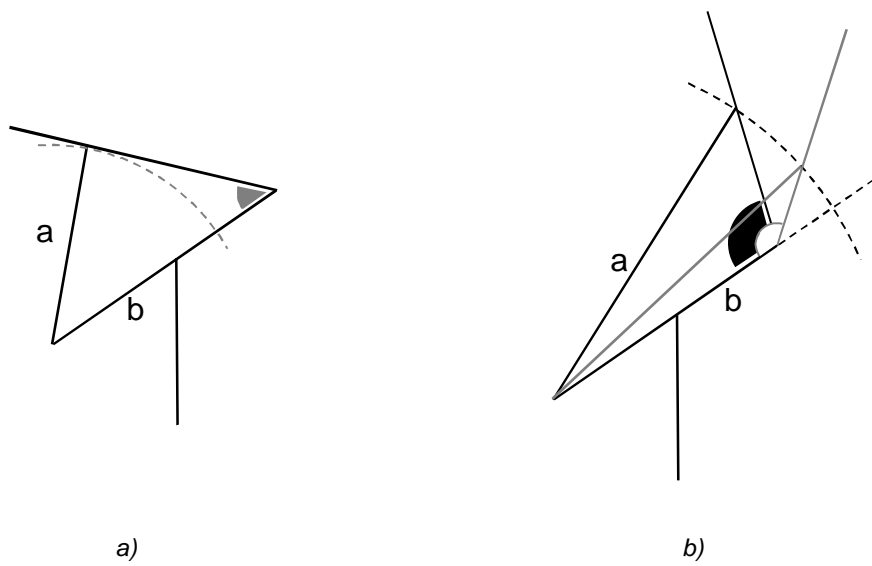


Fig. 30

2. Anejos a la memoria

En el caso en que $a < b$ podemos ver como aunque el ángulo entre a y b aumente, el ángulo sombreado no podrá aumentar más allá de un valor máximo.

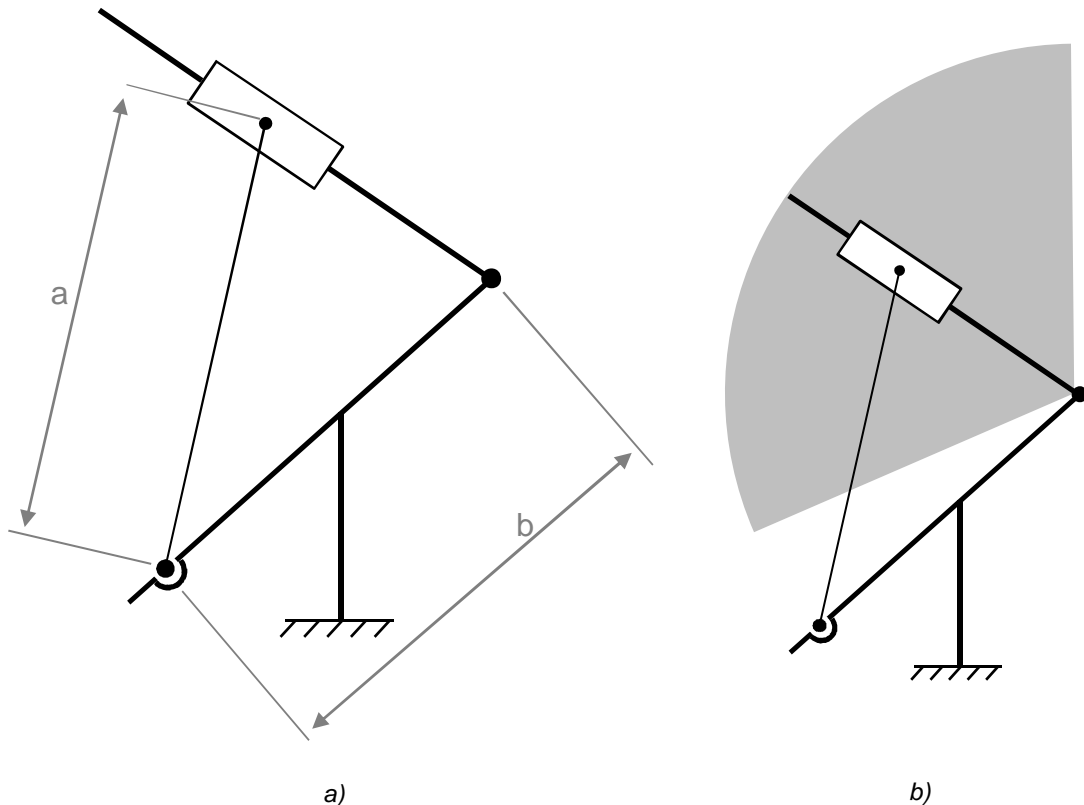


Fig. 31

En cambio en el caso en que $a > b$ si cambiamos el ángulo entre a y b , el ángulo sombreado puede tomar cualquier valor

2.4.2. Selección de los parámetros del mecanismo

Teniendo en cuenta lo extraído en el anterior punto, la barra “ a ” debe ser mayor que la “ b ” para poder alcanzar cualquier posición.

El problema tiene múltiples combinaciones, ya que a , b y θ son magnitudes físicas que una vez sea construido el seguidor no podrán ser variadas.

Por tanto, a continuación vamos a proceder a elegir unos valores de a , b y θ que nos permitan construir un seguidor que funcione correctamente.

Debido a la complejidad del problema, vamos a proceder a resolverlo mediante la simulación y el tanteo.

Usaremos SolidWorks como herramienta de simulación.

2. Anejos a la memoria

Para ello construiremos un modelo que simule el mecanismo anteriormente descrito. Una vez lo tengamos iremos variando las longitudes para obtener una relación de magnitudes satisfactorias.

A partir del estudio de movilidad hemos concluido que la dimensión "a" debe ser mayor a la "b" así que esa condición la deberemos mantener durante todo el proceso.

Además la barra que contiene la deslizadera debe ser capaz de ir desde la posición vertical a situarse con pendiente negativa.

El modelo se ha construido de tal forma que las partes de las piezas que transcurren longitudinalmente junto con las magnitudes "a" y "b" han sido modeladas por extrusión, de tal forma que para modificar su longitud solo hay que seleccionar la pieza y editar la variable de longitud extruida.

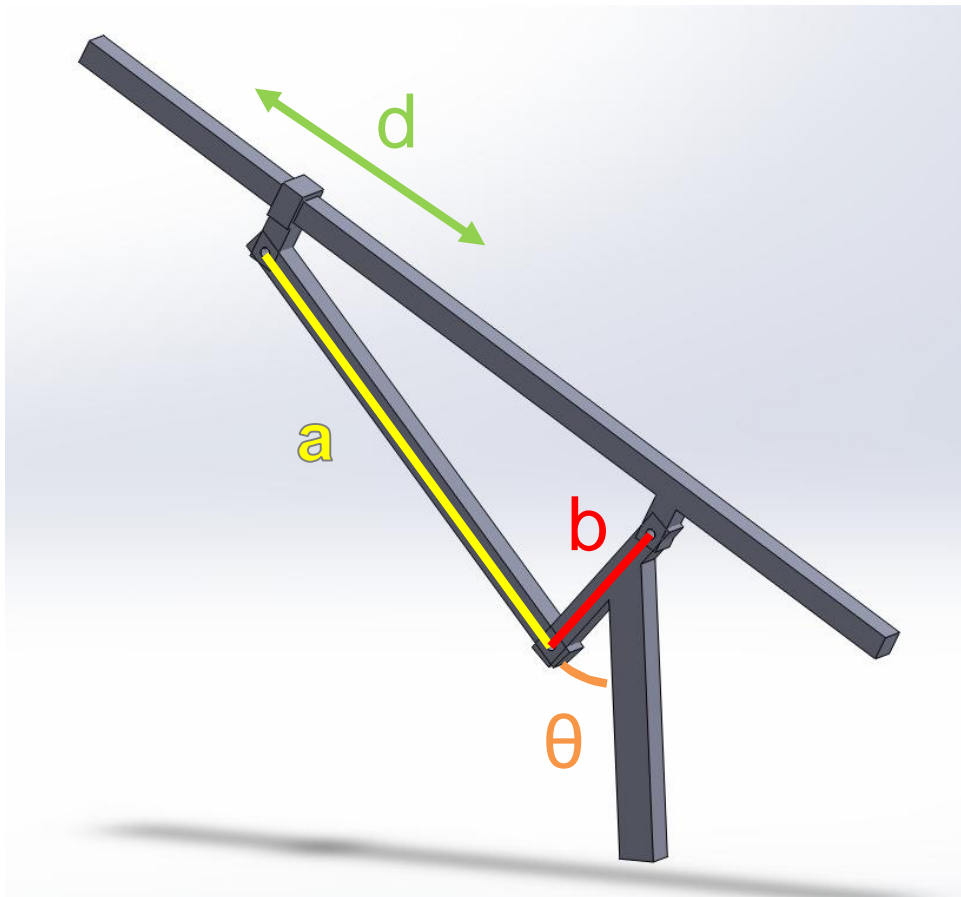


Fig. 32

La combinación seleccionada de magnitudes que cumplen las condiciones impuestas son:

a = 930mm
b=330mm
 $\theta=30^\circ$
d= 541mm

2. Anejos a la memoria

Además, se calculó la variación de posición del seguidor, para saber cuál debe ser la longitud (d) mínima que debe tener la deslizadera.

Nótese que este conjunto de longitudes puede usarse para obtener mecanismos semejantes, de diferente tamaño entre sí pero con la misma relación entre las longitudes.

La distancia entre el final de la barra “a” y el seguidor de la deslizadera no se ha calculado puesto que será reajustado en función de los actuadores seleccionados.

2.5. Parámetros de la transmisión.

Ahora que tenemos definido el mecanismo, necesitamos un sistema que nos permita hacer llegar la potencia de los motores a donde la necesitamos. Vamos a definir algunos parámetros que necesitaremos en los siguientes apartados:

Relación de transmisión, i

Para un sistema de transmisión determinado, se define como el cociente entre la velocidad de entrada a la transmisión y la de salida.

$$i = \frac{v_{salida}}{v_{entrada}}$$

Puede ser adimensional o tener unidades, ya que las velocidades pueden ser lineales o angulares

Ventaja mecánica (VM)

Es la relación entre la fuerza aplicada en la entrada de la transmisión y la obtenida en la salida

$$VM = \frac{Fuerza\ obtenida}{Fuerza\ aportada}$$

Rendimiento de la transmisión, η

Es la relación entre la energía disponible en la salida del sistema y la energía aplicada en la entrada

$$\eta = \frac{E_{salida}}{E_{entrada}}$$

2. Anejos a la memoria

Y en lo que respecta a los motores paso a paso:

Número de pasos del motor

Indica la cantidad de posiciones que un motor toma para realizar una vuelta completa.

Paso angular, α

Se corresponde con el ángulo girado por el motor en cada paso

Por ejemplo, una configuración bastante común son los motores de 200 pasos. Esto significa que si lo hacemos girar 100 pasos habrá dado media vuelta (180°)

Podemos relacionar el número de pasos del motor con el paso angular como:

$$Paso\ angular = \frac{360^\circ}{Núm.\ pasos\ del\ motor}$$

En este ejemplo, el paso angular del motor será de 1.8°

Resolución

En robótica, la resolución de una máquina se corresponde con la menor variación posible de su posición.

La resolución de un motor paso a paso coincide con su paso angular

Vamos a proceder a la delimitación de estos parámetros para los mecanismos de elevación y rotación

2.5.1. Mecanismo de Elevación.

En este caso utilizaremos un sistema de tuerca-tornillo sinfín accionada por un motor paso a paso.

Las características principales de un tornillo sinfín son:

Paso, p

Se define como la distancia entre filetes del tornillo, en un tornillo sinfín se corresponde con el avance de la tuerca por cada vuelta dada por el tornillo. (no confundir con el número de pasos del motor)

Vamos a obtener lo para la transmisión tornillo sinfín-tuerca

Relación de transmisión del sinfín:

$$i_{sinfín} = \frac{Vel.\ Salida}{Vel.\ Entrada} = \frac{p\ (paso\ del\ sinfín)}{360}$$

2. Anejos a la memoria

Relación de transmisión total.

$$i_{elevation} = \frac{v_{tuerca}}{\omega_{motor}} = i_{sinfin} \cdot i_{reductora(elev)}$$

Dónde $i_{reductora(elev)}$ es la relación de transmisión entre la salida del motor paso a paso y el tornillo sinfín.

Tanto $i_{elevation}$ como i_{sinfin} tendrán unidades de longitud/grado, normalmente, mm/° si hemos introducido el paso en milímetros.

Por tanto, si tenemos la tuerca en una posición x_2 , y queremos moverla a una posición x_1 , debemos trasladarnos:

$$x_1 - x_2 = \Delta x$$

Lo que implica que el motor deberá girar:

$$\text{ángulo girado por el motor} = \Delta x \cdot \frac{\omega_{motor}}{v_{tuerca}} = \frac{\Delta x}{i_{elevation}}$$

Este ángulo nos vendrá dado en grados, pero para poder programarlo lo que queremos es obtener el número de pasos. Para ello bastará dividir entre el paso angular, g (°/paso)

$$\text{Número de pasos} = \frac{\Delta x}{i_{elevation} \cdot g}$$

Y la resolución del conjunto de la transmisión será:

$$\text{Resolución} = \Delta x_{min} = g * i_{elevation} = g \frac{p}{360} i_{reductora(elev)}$$

Que no es más que la resolución del stepper multiplicada por la relación de transmisión total.

2.5.2. Mecanismo de Rotación

Este movimiento es bastante más simple, ya que aquí únicamente tendremos el motor y una reductora.

En este caso la relación de transmisión total será igual a la relación de transmisión que tenga la reductora seleccionada.

$$i_{rotacion} = \frac{\omega_{heliostato}}{\omega_{motor}} = i_{reductora(rot)}$$

2. Anejos a la memoria

Y, para obtener el número de pasos a girar para obtener el incremento de rotación deseado:

$$\text{Número de pasos} = \frac{\Delta\alpha}{i_{\text{rotación}} \cdot g}$$

Y análogamente al caso anterior, podemos definir la resolución como:

$$\text{Resolución} = \Delta\alpha_{\text{mín}} = g * i_{\text{rotación}}$$

2.6. Control del mecanismo

En apartados anteriores hemos obtenido todos los parámetros necesarios para poder definir el movimiento del seguidor. A partir de ellos, podemos implementar el programa de control

2.6.1. Microcontrolador

Utilizaremos un microcontrolador Arduino.

Arduino es una compañía de hardware libre y una comunidad tecnológica que diseña y manufactura placas de desarrollo de hardware y software, compuesta respectivamente por circuitos impresos que integran un microcontrolador y un entorno de desarrollo (IDE), en donde se programa cada placa. Arduino se enfoca en acercar y facilitar el uso de la electrónica y programación de sistemas embebidos en proyectos multidisciplinarios. Toda la plataforma, tanto para sus componentes de hardware como de software, son liberados con licencia de código abierto que permite libertad de acceso a ellos.

Estos son los motivos por los cuales hemos elegido este microcontrolador, por la facilidad de programación y por la enorme cantidad de bibliografía, ejemplos e información disponible en la red.

Hemos elegido el Arduino UNO, que se caracteriza por encontrarse en la gama media y tener un precio muy asequible.

2. Anejos a la memoria

Su ficha técnica:

Technical specs

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Los dos aspectos más críticos a la hora de comprobar si el microcontrolador será adecuado son el número de pines y la memoria.

En nuestro caso, sí será suficiente. Puesto que para el programa esencial de control utilizaremos 10 pines digitales. Ya que nuestro sistema se compondrá de dos motores paso a paso (4 pines digitales para cada uno) y dos finales de carrera para controlar la posición (un pin cada uno).

Además el programa ocupa menos de 31.5KB, que es el espacio máximo disponible.

2.6.2. Entorno de programación

Cuando abrimos el programa de Arduino por primera vez o abrimos un archivo nuevo, nos aparece la siguiente ventana:

```
void setup() {  
  // put your setup code here, to run once:  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

Fig. 33

En ella podemos ver:

Un espacio antes del primer bucle. Es dónde se suelen declarar las variables, definir los pines que utilizaremos e incluir las librerías.

Un bucle *setup()*. Es dónde se deben inicializar algunos parámetros necesarios para el correcto funcionamiento del programa.

Un bucle *loop()*. Es dónde se incluye el programa principal, el cuál será ejecutado de forma continua e ininterrumpida, mientras el controlador tenga una alimentación suficiente.

Un espacio después de los bucles. Es dónde se suelen definir las funciones creadas por el usuario, a las cuales podremos llamar desde el diferentes partes del programa.

Aspectos a tener en cuenta.

Arduino diferencia entre mayúsculas y minúsculas. (Pin1 ≠ pin1)

Todas las sentencias deben acabar con un punto y coma (;) sino, el compilador dará error.

Particularidades de Arduino respecto C++

El lenguaje de Arduino se parece en su gran mayoría al lenguaje C++, no obstante debe incluir algunas diferenciaciones que permitan adaptar el lenguaje de programación al microcontrolador.

Concretamente vamos a hablar del proceso de asociación de las entradas y salidas digitales del controlador.

2. Anejos a la memoria

Con la función `#define` podemos asociar un nombre a un pin, que el programa tratará como si se tratase de una variable. La sintaxis es la siguiente:

```
#define nombre_del_pin numero_del_pin
```

Por ejemplo:

```
#define N1 13;
```

Esta sentencia asigna el nombre "N1" pin 13 de nuestro Arduino. Y debe aparecer antes de los bucles.

Una vez hecho esto, hay que definir si el pin es una entrada o una salida. Para ello usamos la función `pinMode`

```
pinMode(nombre_del_pin, INPUT/OUTPUT);
```

Por ejemplo:

```
pinMode(N1, OUTPUT);
```

De este modo hacemos que el pin 13 sea una salida digital. Esto debe ser incluido en el bucle `setup()`.

Para poder controlar nuestros actuadores, necesitaremos hacer modificaciones en la variable N1, para que en el pin 13 aparezcan 0V o 5V. Para ello usamos la función `digitalWrite`

```
digitalWrite(nombre_del_pin, HIGH/LOW);
```

Por ejemplo

```
digitalWrite(N1, HIGH);
```

Cuando el programa llegue a esta sentencia, el pin 13 habrá 5V.

En el caso de tener una salida digital:

```
#define N2 12;  
pinMode(N2, INPUT);
```

En algún punto del programa necesitaremos comprobar su valor, para ello utilizaremos la función `digitalRead`

```
digitalRead(nombre_del_pin);
```

Por ejemplo

2. Anejos a la memoria

`digitalRead(N2);`

Esta función devolvería un valor booleano *HIGH* (1) o *LOW* (0)

Finalmente hablaremos de las funciones utilizadas para comunicar el Arduino con el ordenador.

Para ello utilizaremos las funciones:

`Serial.begin(velocidad);`

Esta función es la que se regula el número de bits por segundo que circulan entre el controlador y el PC. El requisito para que funcione es que el valor de la velocidad esté entre 300 – 115200 y que el la velocidad del Arduino y del programa Arduino en el ordenador sean la misma. Se suele tomar la que aparece por defecto, 9600 baudio.

`Serial.print(variable);`

Está función sea llamada nos mostrará por el Monitor Serie el valor de la variable.

El Monitor Serie puede ser abierto desde el menú Herramientas/Monitor Serie o con la combinación de teclas 'Ctr + Mayus + M'

Desde el Monitor Serie también podemos ver y modificar el valor de bits por segundo

Con estas dos funciones podemos realizar un importante trabajo de corrección de errores sin poner en peligro la parte mecánica conectada al controlador.

2.6.3. Librerías y funciones *built in* utilizadas

Lo visto hasta ahora son funciones que Arduino incluye por defecto. No obstante para nuestra aplicación necesitaremos incluir algunas funciones más. Las más importantes serán las incluidas en la librería Time.h

La extensión de las librerías en Arduino es ".h" y la forma de incluirlas es:

```
#include<Time.h>
```

Aunque para que funcione necesitamos habernos descargado el archivo correspondiente y ponerlo dentro de la carpeta correspondiente en los archivos del programa.

La librería Time.h permite trabajar con variables que miden el tiempo. Dichas variables son de tipo time_t

2. Anejos a la memoria

Si queremos que “tiempo” sea una variable de este tipo debemos declararla como:

```
time_t tiempo;
```

Las variables tipo `time_t` almacenan el tiempo en modo de segundos. Por tanto si queremos obtener la variable `t'` que se corresponde con el día siguiente de `t` será: $t' = t + (1\text{día} * 24\text{horas} * 60\text{minutos} * 60\text{segundos}) = t + 86400$

Las funciones predefinidas (*built in*, no definidas por el usuario) que más utilizaremos son:

`setTime`: Sirve para poner en hora el sistema. Sus argumentos deben ser:

```
setTime(hora, minutos, segundos, días, mes, año)
```

`now()`: devuelve una variable `time_t` cuyo valor es el correspondiente al instante en que la función ha sido llamada.

Funciones para leer parámetros temporales:

Función	Descripción	Rango de valores
<code>hour(tiempo)</code>	Devuelve la hora correspondiente a la variable “tiempo”	0-23
<code>minute(tiempo)</code>	Devuelve los minutos correspondientes a la variable “tiempo”	0-59
<code>second(tiempo)</code>	Devuelve los segundos correspondientes a la variable “tiempo”	0-59
<code>day(tiempo)</code>	Devuelve el día correspondiente a la variable “tiempo”	0-31
<code>month(tiempo)</code>	Devuelve el mes correspondiente a la variable “tiempo”	0-12
<code>year(tiempo)</code>	Devuelve el año correspondiente a la variable “tiempo”	A partir de 1970

Por defecto, Arduino no es capaz de trabajar con fechas anteriores al 1 de Enero de 1970.

2.7. Descripción del Programa

Vamos a explicar el programa que hemos desarrollado para controlar el seguidor:

Los pasos (simplificados) que llevará a cabo el autómata para realizar el seguimiento del Sol serán los siguientes:

- 1- Cálculo de la hora y azimut de la salida y puesta del Sol
- 2- Colocación en la posición correspondiente a la salida del Sol
- 3- A partir de la salida del Sol, ir recalculando y posicionándose en su próxima posición.
- 4- Una vez llegada la hora de puesta del Sol, detención de la rutina 3 y vuelta al punto 1.

2.7.1. Constantes necesarias:

Relativas al punto de instalación:

phi: latitud, ϕ

L: longitud del punto, λ

Lm: longitud del huso horario, λ_m

Relativas al mecanismo y la transmisión

barr_est: es la longitud de la barra estática, en mm

barr_movil: longitud de la barra móvil, en mm

tect: es la inclinación de la barra estática, en $^\circ$

En nuestro proyecto han sido definidas como 330mm, 930mm y 60°

paso_sf: es el paso del sinfín, en mm

ir_elev: es la reducción entre la salida del stepper y el tornillo
en formato decimal ($1/4 = 0.25$)

g_elev: es el paso angular del stepper de elevación ($^\circ/\text{paso}$)

i_rot: es la reducción en el motor de rotación

en formato decimal (en este proyecto será necesaria de unos $1/200 = 0.005$)

g_rot: paso angular motor de rotación ($^\circ/\text{paso}$)

al_fc: altitud solar al inicio de día, γ_s

Se corresponde con la inclinación del Sol asociada a la posición del seguidor cuando esta elevado al máximo (final de carrera activado)

2. Anejos a la memoria

az_fc: acimut al inicio del día.

Se corresponde con la posición del final de carrera medida respecto al este. Se recomienda que sea 0° (final de carrera situado en el Este)

t_V, t_H: definen el tiempo de espera entre pasos del motor, lo cual establece la velocidad de giro del motor

2.7.2. Variables necesarias:

delta: declinación, δ

dx, dy, dz: variables auxiliares para el cálculo de delta

h: variable en la que se irá almacenando y machacando el ángulo horario a lo largo del día

hs, hp: ángulo horario de salida y puesta del Sol

TSV: variable en la que se irá almacenando y machacando el TSV a lo largo del día

TSVs, TSVp: hora solar verdadera de salida y puesta del Sol

HOp, HOs: hora oficial de puesta y salida del Sol

ET: Ecuación del tiempo, ET

B: variable auxiliar de ET

azs: acimut solar, ψ_s

azss: acimut solar de salida del Sol

azss_next: acimut solar inmediatamente siguiente a azss

flag: nos indicará si el año actual es bisiesto o no

fin_bisiesto: si el año actual es bisiesto, tendrá el valor del año que viene

fin_día: la usaremos para marcar que el día ha terminado (se ha puesto el Sol)

crepusculo: si vale 1 es que la altitud solar es mayor al ángulo de crepúsculo

mediodia: controla si se ha alcanzado el mediodía solar

día_actual: será necesaria para saber que ha llegado el día siguiente.

StepsH, StepsV: llevan la cuenta del paso en el que se encuentra cada stepper.

Toma valores de 0 a 3 si el tipo de control es a pasos completos y de 0 a 7 si es a medios pasos

2. Anejos a la memoria

DirectionH, DirectionV: controlan la dirección de giro de los steppers

Variable	Valor	Significado
DirectionH	1	Giro Este → Oeste
	0	Giro Oeste → Este
DirectionV	1	Descenso de las placas
	0	Elevación de las placas

Paso: es la matriz que se controla el orden de excitación de las bobinas de los steppers

n_al, n_az: número de pasos que debe dar el motor de elevación (al) o rotación (az) para moverse a la próxima posición.

FC_V, FC_H: se corresponden con los finales de carrera acimutal y de elevación.

2.7.3. Funciones necesarias:

HO(now()): La función "Hora Oficial" devuelve la hora en el momento que es ejecutada, en formato decimal (las 11:45 son las 11.75)

HO(t): siendo t una variable tiempo, nos devuelve la "Hora Oficial" asociada a dicha variable

z(): La función día devuelve el día actual, en formato 1 – 365

sind, cosd: funciones seno y coseno, con el argumento en grados

asind, acosd: funciones arco-seno y arco-coseno en grados

als(hora, día): devuelve la altitud solar correspondiente al día y hora introducidos.

azs(hora, día): devuelve la acimut solar correspondiente al día y hora introducidos.

stepperH, stepperV: cada vez que son llamadas avanzan un paso del motor acimutal (H) o de elevación (V)

DirectionH, DirectionV: controlan el sentido de giro de su motor

SetDirectionH, SetDirectionV: Función auxiliar para llevar la cuenta del paso en que se encuentra el motor.

elevacion(A,B): dadas dos elevaciones solares A y B, devuelve el número de pasos necesarios para ir de una posición a otra.

rotacion(A,B): dados dos ángulos acimutales A y B, devuelve el número de pasos necesarios para ir de un ángulo a otro.

pararV(), pararH(): deben ser llamadas después de haber completado el movimiento de su respectivo motor, para evitar un consumo de energía excesivo.

2.7.4. Resumen del programa.

Al final del día, el seguidor se coloca en la posición en la que saldrá el Sol al día siguiente.

Un bucle espera a que empiece el día. Es decir, que llegue el día siguiente y a la hora de salida del Sol.

En cuanto empieza el día, se comprueba si la elevación solar es mayor al crepúsculo.

Si no lo es, se dejará el motor de elevación tal y como está y se irá colocando el seguidor en la posición acimutal correspondiente. Y se repite el bucle.

Cuando la altitud Solar supera al crepúsculo, a la colocación acimutal se une la variación de inclinación de las placas, iniciando el seguimiento total.

El seguimiento se lleva a cabo de forma diferencial entre la posición que tiene el seguidor en el momento del cálculo y la posición calculada. Hay una función que calcula el número de pasos a dar entre ambas posiciones. Una vez el seguidor está en la nueva posición, cesa la alimentación para reducir el consumo.

Una vez se llega al mediodía, la velocidad del motor de elevación se invierte, para que la inclinación de las placas deje de descender y comience a ascender.

Esta es la rutina diaria. Dentro de la cual hay un bucle cuya función es detectar el fin del día. Esto ocurre cuando llega la hora de puesta del Sol.

Llegados a este punto, se re-calculan todos los parámetros necesarios para definir el punto de salida del Sol del día siguiente y se coloca al seguidor dicha posición.

La rutina completa ha sido ejecutada, y el microcontrolador vuelve a esperar a que llegue el nuevo día.

2. Anejos a la memoria

A continuación explicamos más detalladamente los bucles principales:

Bucle principal:

```
while (fin_dia ==0
{
    // rutina principal del día

    if (HO(>HOp){
        fin_dia=1;

        //obtenemos dia_actual y recalculamos HOP y HOs para
el día siguiente

        }
    }

if (fin_dia ==1)
{
    if (z()==dia_actual && HO(>HOs) //esperamos al día
siguiente
    {
        fin_dia=0;
    }
}
```

La variable `fin_dia` es la que nos marca el estado en el que se encuentra el seguidor.

Mientras `fin_dia` es igual a 0, se ejecuta el bucle *while*:

```
while (fin_dia ==0
{
    // rutina principal del día

    if (HO(>HOp){
        fin_dia=1;

        //obtenemos dia_actual y recalculamos HOP y HOs para
el día siguiente

        }
    }
}
```

2. Anejos a la memoria

En *//rutina principal del día* irá el programa que se encargará de ir calculando la posición solar tras cada intervalo de tiempo, y colocando el seguidor en la posición correspondiente.

La salida del bucle se produce cuando se pone el Sol, esto es, cuando la hora actual, HO() es mayor a la hora de puesta del Sol, HOp.

Llegados a este punto, la variable fin_día pasa a valer 1.

Calculamos la z() para el día siguiente, y asignamos su valor a dia_actual

Además, calculamos las horas de salida y puesta del Sol para dia_actual y asignamos sus valores a HOs y HOp.

Una vez hecho esto, con el nuevo valor de fin_día entramos al bucle *if*:

```
if (fin_dia ==1)
{
  if (z()==dia_actual && HO(>HOs) //esperamos al día
siguiente
  {
    fin_dia=0;
  }
}
```

Dentro de este bucle hay otro bucle *if* que espera a que llegue el día siguiente (día actual dado por z() igual a la variable dia_actual) y a que sea la hora de salida del sol (hora actual, HO() mayor que la hora de salida del Sol, HOs)

Entonces, la variable fin_día pasa a valer 0 y el bucle while inicial vuelve a ejecutarse.

Obtención de las horas de salida y puesta del Sol.

Lo primero que debemos hacer es calcular las horas de salida y puesta del Sol. Puesto que estas dos variables serán las que controlarán las líneas generales del programa, y por tanto serán las que aparecerán en el bucle principal del programa

Sabemos que a la hora del amanecer y anochecer la altitud solar es, por definición, igual a 0. Por tanto, a partir de la ecuación de la altitud, obtenemos:

$$0 = \sin(\phi) \cdot \sin(\delta) + \cos(\phi) \cdot \cos(\delta) \cdot \cos(h)$$

2. Anejos a la memoria

En la que ϕ (latitud) es constante todo el tiempo, y δ (declinación) depende directamente del día del año. Por tanto, calculamos la declinación cómo:

$$\delta = 23,45 \cdot \sin\left[360 \cdot \frac{284 + z}{365}\right]$$

En el programa:

```
dx = 360L*(284+z());  
dy = 365;  
dz=dx/dy;
```

```
delta = 23,45*sind(dz);
```

El motivo por el que esta variable se calcula de forma separada es para evitar que la operación se desborde.

Por ese motivo, declaramos dx como "long" y añadimos la letra "L" junto al 360 para que trate los dígitos como tal.

Por tanto podemos obtener el ángulo horario (h) como:

$$h = \cos^{-1}\left(\frac{-\sin(\phi) \cdot \sin(\delta)}{\cos(\phi) \cdot \cos(\delta)}\right)$$

De aquí obtendremos dos valores para h, uno comprendido entre 0° - 180° y el otro comprendido entre 180° - 360° cada uno de ellos estará asociado con la salida o puesta del Sol.

Ahora que ya tenemos "delta" y "phi" es una constante, obtenemos los dos ángulos horarios:

```
hp = acosd((-sind(phi)*sind(delta))/(cosd(phi)*cosd(delta)));  
hs = -hp;
```

Una vez obtenido h, lo usaremos para calcular las dos horas solares, como:

$$TSV = 12 - \frac{h}{15}$$

Lo escribimos como:

```
TSVp = 12+(hp/15);  
TSVs = 12+(hs/15);
```

Ahora necesitaremos calcular la ecuación del tiempo:

$$ET = 9,87 \cdot \sin 2B - 7,53 \cos B - 1,5 \sin B$$

2. Anejos a la memoria

$$\text{siendo } B = \frac{360}{364}(z - 81)$$

Lo implementamos:

$$B = (360/364) * (z()-81);$$

$$ET = 8,87*\text{sin}(2*B) - 7,53 * \text{cos}(2*B) - 1,5*\text{sin}(B);$$

Finalmente con las horas solares y la ecuación del tiempo obtenemos la hora Oficial de salida y puesta del Sol:

$$HO = TSV + \varepsilon - ET + \frac{\lambda - \lambda_m}{15}$$

Cabe recordar que ambos TSV los acabamos de calcular, ε y ET dependen únicamente del día del año (y no de la hora, por tanto es la misma para ambas) y los λ dependen de la longitud geográfica del punto de instalación.

$$HO_p = TSV_p - (ET/60) + (L - L_m)/15;$$

$$HO_s = TSV_s - (ET/60) + (L - L_m)/15;$$

Nótese que el valor de ET se encuentra en minutos, por tanto hemos de dividir entre 60 para pasarlo a horas en formato decimal

Llegados a este punto hemos calculado: HO_{puesta} , HO_{salida}

Vamos a ver cómo utilizamos las variables, en el bucle que controla el paso de los días:

Cálculo de los ángulos de salida del Sol

A continuación, necesitamos calcular el ángulo azimutal de salida y puesta del Sol:

$$\cos(\psi_s) = \frac{\cos(\delta) \cdot \sin(h)}{\cos(\gamma_s)} = \frac{\cos(\delta) \cdot \sin(h)}{\cos(0)} = \cos(\delta) \cdot \sin(h)$$

Lo obtendremos a partir de magnitudes que previamente hemos calculado, concretamente de la declinación y el ángulo horario.

Cabe recordar que tendremos dos ángulos horarios, por tanto habrá que utilizar el adecuado para que el que calculemos sea el de salida y no el de puesta.

No obstante, tenemos el problema que la función arc-coseno devuelve un valor de ángulo entre 0° - 180° y hay días del año en los que el recorrido acimutal del Sol es mayor de 180°

2. Anejos a la memoria

Hemos calculado el rango de valores de Acimut y Altitud Solar para $z = 100$, el 10 de Abril de un año no bisiesto.

Obtenemos la siguiente gráfica:

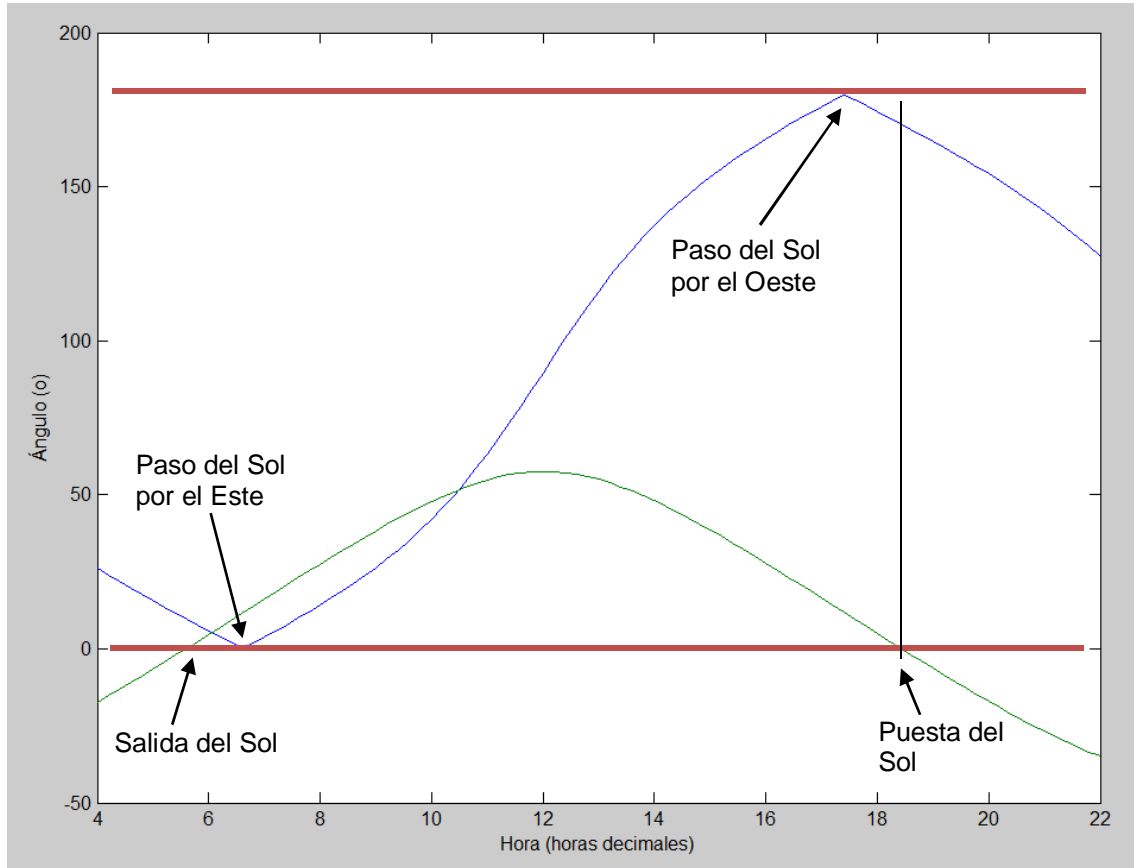


Fig. 34

Dónde podemos ver en el eje horizontal las horas, en formato decimal y en el vertical el ángulo acimutal (azul) y la altitud solar (verde)

Hay que tener en cuenta que a pesar de que las gráficas están en el mismo plano, en la realidad el ángulo acimutal está contenido en el plano horizontal, y la altitud en un plano vertical perpendicular a él.

2. Anejos a la memoria

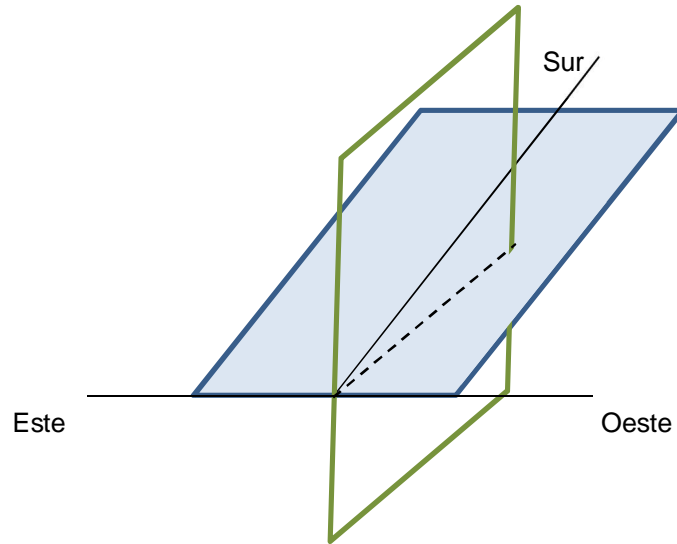


Fig. 35

El plano azul es el que contiene el ángulo acimutal en todo momento. El plano del borde verde, contiene la altitud solar y es perpendicular en todo momento al azul. Su ángulo respecto la línea del Sur va cambiando, y dicho ángulo es el Acimutal.

Si volvemos a la gráfica, se puede apreciar como el origen de ángulos acimutales se encuentra en el este geográfico. Por tanto nuestro programa nos devolverá ángulos acimutales comprendidos en el rango:

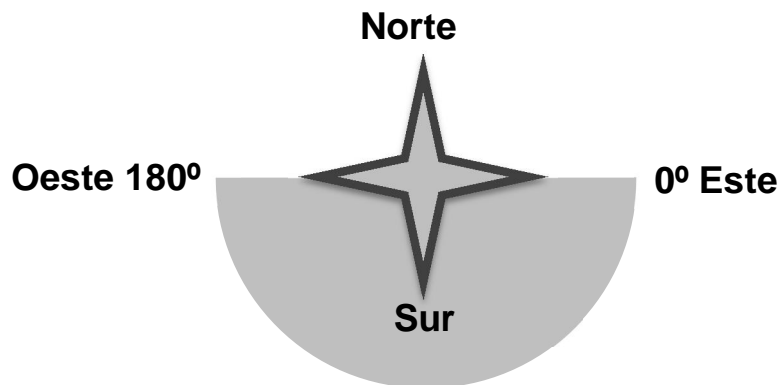


Fig. 36

2. Anejos a la memoria

Vamos a representar sobre este esquema, el movimiento acimutal real del Sol, y los valores que nos daría el programa:

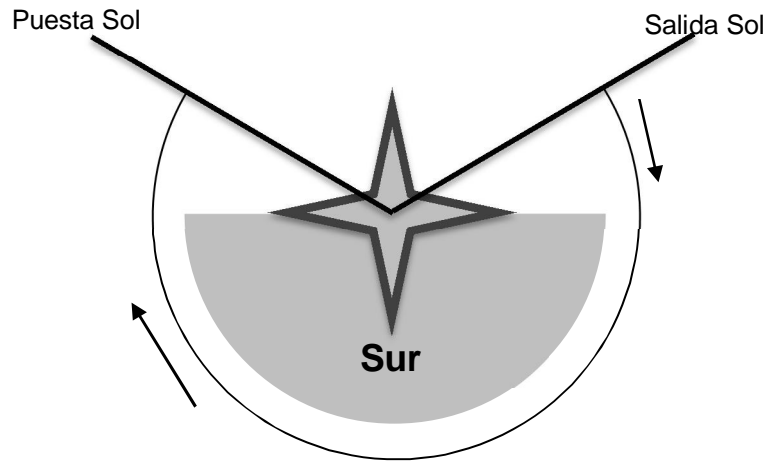


Fig. 37 Movimiento real del Sol

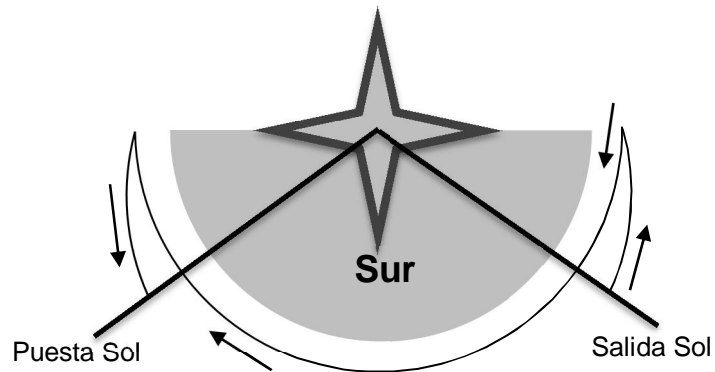


Fig. 38 Valores de ángulo devueltos por el programa

Sabemos que el ángulo acimutal se mide respecto al Este.

El problema que tenemos es que una vez obtenido el ángulo de salida del Sol, no podemos saber si dicho ángulo se encuentra en el Noroeste o en el Sudeste:

2. Anejos a la memoria

Ejemplo: Si nuestro ángulo de salida de Sol para cierto día es de 30° , no podemos saber dónde nos encontramos

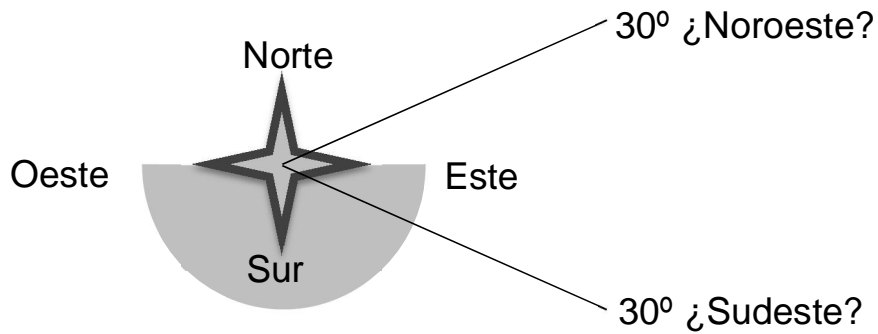


Fig. 39

Una vez calculado el ángulo de salida del Sol (a partir de la hora de salida del Sol), calcularemos el valor de ángulo acimutal para unos minutos tras la salida del Sol.

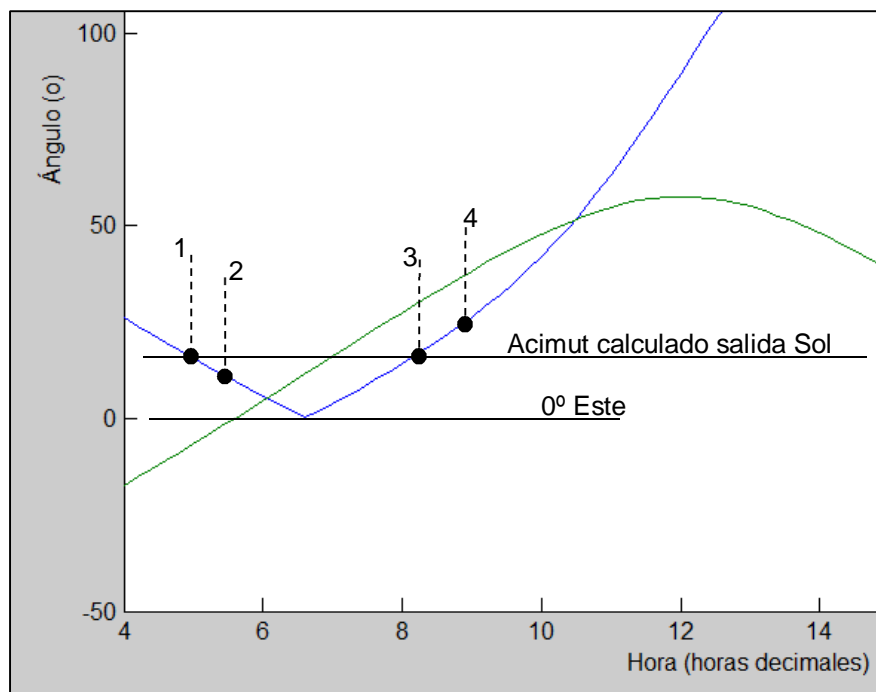


Fig. 40

Ejemplo: Una vez tenemos el acimut de salida, podría ser que estuviéramos en los puntos 1 o 3.

2. Anejos a la memoria

Cómo el ángulo acimutal es siempre creciente a lo largo del día, podemos saber en qué posición estamos con una comparación: (Figura 40)

Si el nuevo valor es mayor al ángulo de salida: significa que el nuevo valor se corresponde con el punto 4 y por tanto nos estamos alejando del este. Estamos en el sudeste.

Si el nuevo valor es menor al ángulo de salida: significa que el nuevo valor se corresponde con el punto 2 y por tanto, nos estamos acercando al Este. Estamos en el noroeste.

Para llevarlo a cabo, hemos comprobado experimentalmente que un incremento de tiempo de unos 36 segundos (0,01 horas) proporciona una variación del ángulo acimutal de unos $0,1^\circ$ lo suficiente como para que el microcontrolador pueda comprobar en que punto nos encontramos.

Para obtener el ángulo de elevación en el momento la salida del Sol es bastante más sencillo. Ya que sabemos que en el instante del amanecer el Sol estaría una altitud 0° , lo que implicaría que las placas estuvieran a 90°

Debido a que cuando el Sol esta a 0° es prácticamente imposible que le llegue luz al seguidor, hemos tomado como valor de referencia el crepúsculo civil, que se corresponde con una altitud solar de unos 10°

Sabiendo esto ya podemos mover el seguidor hasta la posición de salida del Sol y el bucle vuelve a empezar.

Por tanto, cada mañana las placas estarán situadas a una inclinación de 80° , y una vez que la altitud del Sol haya superado los 10° el programa empezara a recalcular la inclinación de las placas.

2.7.5. Explicación de las funciones utilizadas.

Funciones para la gestión del tiempo

HO()

Esta función devuelve la hora en formato decimal en el instante en que es llamada.

Su código es:

```
double HO(time_t)
{
    double result, horas, mints, segs;

    horas = hour(t);
    mints = minute(t);
    segs = second(t);

    result = horas + mints/60 + segs/3600;
    return result;
}
```

En la entrada de la función asignamos a la variable t el tiempo en el instante en que es llamada la función. Esto lo hacemos para evitar resultados inconsistentes.

A continuación obtenemos el valor de la hora, minutos y segundos.

Para obtener el valor de la hora en formato decimal pasamos los minutos y segundos a horas y lo sumamos todo.

Así, las diez menos cuarto y 30 segundos (9:45:30) serían las:

9 horas

45 min/60 = 0,75 horas

30seg/3600 = 0,00833 horas

La hora en formato decimal sería aproximadamente 9,76

Finalmente ese valor es el devuelto por la función.

2. Anejos a la memoria

z()

Cuando esta función es llamada devuelve el día del año, como un valor ente 1 y 366

Su código es:

```
int z()
{
    time_t t = now();

    int days[13]={0,0,31,59,90,120,151,181,212,243,273,304,334};

    int result = day(t) + days[month(t)];

    //Gestión de los años bisiestos:

    if (day(t)==29 && month(t)==2)
    {
        flag = 1;
        fin_bisiesto = year(t)+1;    o
    }

    if (fin_bisiesto == year(t))
    {
        flag = 0;
        fin_bisiesto = 1900;
    }

    if (flag == 1 && month(t)>2)
    {
        result = result + 1;
    }

    return result;
}
```

2. Anejos a la memoria

Igual que en el caso anterior, evitamos resultados inconsistentes asignando el tiempo actual a la variable t.

Esta función nos devuelve el día en el momento que es llamada. No obstante, para algunas aplicación necesitaremos obtener el día z() para un día dado.

Por tanto modificaremos el código:

```
int z()  
{  
    time_t t = now();
```

Y lo reescribiremos como:

```
int z(time_t t)  
{  
    time_t t = now(); //Esta línea la eliminamos
```

De este modo ahora z(now()) es equivalente a la anterior z()

La forma en la que calcularemos el día del año en formato 1 – 366 la realizaremos será la siguiente:

Dado el día y el mes, sumaremos al día el número de días de los meses anteriores.

Por ejemplo, para obtener z() del día 3 de Marzo, a 3 le sumaremos el número de días de Enero y Febrero.

Para ello, definiremos un vector que contendrá la suma de días acumulada para cada mes. Además, la posición de cada cifra coincidirá con su mes.

Mes	Días	Días acumulados	Elemento del vector
Enero	31	0	1
Febrero	28	31	2
Marzo	31	31 + 28 = 59	3
Abril	30	59 + 31 = 90	4
Mayo	31	90 + 30 = 120	5
...etc			

Por ejemplo, a Enero le corresponden 0 días, pues el elemento 1 del vector será 0:

```
days[1] = 0
```

2. Anejos a la memoria

A Febrero le corresponden 31 días, pues el elemento 2 del vector será 31.

```
days[2] = 31
```

Y así sucesivamente.

Aunque debemos tener en cuenta que para Arduino el primer elemento de un vector tiene el índice 0.

Por eso añadimos un 0 al inicio del vector

Con lo que obtenemos el vector:

```
days[13] = {0, 0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334};
```

A continuación, iremos guardando el resultado en la variable result.

De momento valdrá el día actual más el número de días acumulados debidos al mes del año. Lo podemos escribir así:

```
result = day(t) + days[month(t)];
```

A continuación vamos a gestionar los días del año los días bisiestos.

La siguiente sucesión de bucles es válida únicamente para esta aplicación, debido a que su correcto funcionamiento esta ligado a que la función sea llamada todos los días del año.

El control lo realizaremos a partir de tres bucles *if*

Y con la ayuda de las variables:

flag: cuando valga 1 significará que es un año bisiesto

fin_bisiesto: indicará el año siguiente al bisiesto

Es importante visualizar que, el hecho que un año sea bisiesto afecta en que ese año los días a partir del 29 de Febrero, su z() será la misma que cualquier año no bisiesto mas uno.

Lo que hacemos con el primer bucle es comprobar si ese día tiene 29 de Febrero.

Cuando la comparación sea cierta, significará que ese año es bisiesto, por tanto flag pasará a valer 1. Y fin_bisiesto será el año actual más uno.

2. Anejos a la memoria

De aquí el programa pasará al tercer bucle, en el que mientras flag sea igual a 1, sumará 1 al valor de z() calculado anteriormente.

Finalmente hemos incluido otro bucle que cuando llegue el año siguiente volverá a asignar a flag el valor de 0, y a fin_bisiesto el valor 1900 para que no hayan problemas.

Es importante poner este bucle en segunda posición, antes del bucle que suma 1 para evitar que el 1 de Enero del año siguiente a un año bisiesto devuelva el valor 367 en lugar de 1.

Cabe añadir que, como aparecerá en las instrucciones de inicialización si la instalación del seguidor se realiza a partir de Febrero en un año bisiesto se deberán inicializar las variables flag y fin_bisiesto

Funciones trigonométricas.

A nivel operativo, necesitamos operaciones trigonométricas que trabajen con grados. Ya que en Arduino las funciones que hay por defecto trabajan en radianes, vamos a definirnos una serie de funciones que funcionen en grados.

Además, utilizaremos el mismo nombre que estas funciones tienen en Matlab para facilitar la detección de errores.

Estas funciones serán el coseno, seno, arcoseno y arcseno , definidas como sind, cosd, asind y acosd.

En las funciones sind y cosd lo que haremos será multiplicar los argumentos por pi/180. Estos son sus códigos:

Seno de un ángulo en grados:

```
double sind(double x){
  double result;
  const double pi = 3.1415926535;
  result = sin(x*pi/180);
  return result;
}
```

Coseno de un ángulo en grados:

```
double cosd(double x){
  double result;
  const double pi = 3.1415926535;
  result = cos(x*pi/180);
  return result;
}
```

2. Anejos a la memoria

En las funciones `asind` y `acosd` lo que haremos será multiplicar el resultado de la operación por $\pi/180$. Estos son sus códigos:

Arco-seno en grados de un ángulo:

```
double asind(double x){  
  
    const double pi = 3.1415926535;  
    double result;  
  
    result = asin(x)*180/pi;  
    return result;  
  
}
```

Arco-coseno en grados de un ángulo:

```
double acosd(double x){  
  
    const double pi = 3.1415926535;  
    double result;  
  
    result = acos(x)*180/pi;  
    return result;  
  
}
```

Funciones de cálculo de parámetros solares

Ahora, vamos a analizar las funciones que utilizaremos para ir recalculando la posición del Sol a lo largo del día. El proceso de cálculo de dichas variables es el que aparece en el punto **1.5 Definiciones y abreviaturas**, apartado Explicación del movimiento solar y definición de sus variables.

als(hora, día): calcula la altitud solar un día y a una hora dadas. Su código es:

```
double als(double hora, int dia){  
  
    double result;  
  
    dx = 360L*(284+dia);  
    dy = 365;  
    dz=dx/dy;  
  
    delta = 23.45*sind(dz);  
  
    B = (360/364) * (dia-81);  
  
    ET = 8.87*sind(2*B) - 7.53 * cosd(2*B) - 1.5*sind(B);  
  
    //TSV = HO() +ET/60 + (Lm-L)/15;  
  
    TSV = hora +ET/60 + (Lm-L)/15; /  
    h= 15*(12-TSV); //ángulo horario  
  
    result=asind(sind(phi)*sind(delta)+cosd(phi)  
                *cosd(delta)*cosd(h));  
  
    return result;  
}
```

La función, a partir del día y la hora calcula la altitud aplicando las ecuaciones de posición solar descritas. Durante el proceso de cálculo, la función llama a muchas de las funciones descritas en anteriormente.

Cabe indicar que volvemos a utilizar las variables auxiliares dx, dy, dz para evitar el desbordamiento de la división durante en cálculo de delta.

2. Anejos a la memoria

Finalmente, la función devuelve el valor de la altitud solar en formato decimal.

`azs()` calcula el ángulo acimutal un día y a una hora dada. Su código es:

```
double azs (double hora, int dia){
    double result;
    double altitud;
    altitud = als(hora, dia);
    result= acosd((cosd(delta)*sind(h))/cosd(altitud));
    return result;
}
```

Puesto que el ángulo acimutal está relacionado con la altitud solar, llamamos a la función anterior y usamos su resultado para esta función.

Cabe destacar que debemos calcular la altitud antes de calcular el resultado. Esto se debe a que las variables `delta` y `h` son variables globales que la función “`als`” puede modificar. Por eso utilizamos la variable local `altitud` y no hemos escrito directamente:

```
result= acosd((cosd(delta)*sind(h))/cosd(als(hora, día)));
```

Ya que en este caso los valores de `delta` y `h` no tendrían porqué ser los correctos.

Activación de los motores paso a paso

Para llevar a cabo el control de los motores, realizaremos una función que cada vez que sea llamada moverá el motor ciertos pasos (paso completo o medio paso en función de la configuración). Puesto que tendremos dos motores, haremos una función para cada motor.

Estas funciones se diferencian de las anteriores en que necesitaremos que utilicen y modifiquen variables globales. Así como que definamos ciertos elementos fuera del cuerpo de la función de uso explícito para ellas, cosa que en el resto de funciones no pasaba.

Las funciones serán `stepperH()` para el actuador acimutal y `stepperV()` para el que controlará la inclinación.

Además cada uno de ellos ira acompañado de otra función que controlará la dirección y el paso en que se ha dejado el motor.

Estas funciones serán `SetDirectonH()` y `SetDirectionV()`

El sentido de giro lo controlaremos con una variable de tipo booleano, la cuál hará que gire en un sentido cuando valga 1 y que gire en sentido contrario cuando valga 0.

2. Anejos a la memoria

El paso en el que se encuentra el motor lo controlaremos con una variable de tipo número entero

Configuración previa

Para controlar un motor paso a paso necesitaremos ir excitando secuencialmente las bobinas que lo componen.

Para llevarlo a cabo asociaremos una salida digital para cada bobina:

Para los pines de control del stepper Horizontal:

```
#define IN1 12
#define IN2 11
#define IN3 10
#define IN4 9
```

Para los pines de control del stepper Vertical:

```
#define IN5 8
#define IN6 7
#define IN7 6
#define IN8 5
```

Y en el bucle *setup()* tendremos que poner que queremos que estos pines sean tratados como salidas digitales.

```
pinMode(IN1, OUTPUT);
pinMode(IN2, OUTPUT);
pinMode(IN3, OUTPUT);
pinMode(IN4, OUTPUT);
pinMode(IN5, OUTPUT);
pinMode(IN6, OUTPUT);
pinMode(IN7, OUTPUT);
pinMode(IN8, OUTPUT);
```

A continuación debemos definir el “orden” en que iremos excitando cada bobina. Para ello definiremos una matriz llamada *Paso* que incluye la secuencia de excitación de las bobinas para un control de medio paso:

```
int Paso [ 8 ][ 4 ] =
{
  {1, 0, 0, 0},
  {1, 1, 0, 0},
  {0, 1, 0, 0},
  {0, 1, 1, 0},
  {0, 0, 1, 0},
  {0, 0, 1, 1},
  {0, 0, 0, 1},
  {1, 0, 0, 1}
};
```

*Nota** El control de un motor paso a paso mediante medios pasos nos proporciona una precisión de movimiento mayor, y un consumo y par intermedio.

2. Anejos a la memoria

No obstante puede que en nuestro seguidor este modo no nos proporcione el par necesario o nos de problemas. Entonces deberemos controlarlo mediante pasos completos

En ese caso, deberemos utilizar la matriz:

```
int Paso [ 4 ][ 4 ] =
{
    {1, 1, 0, 0},
    {0, 1, 1, 0},
    {0, 0, 1, 1},
    {1, 0, 0, 1}
};
```

La forma más rápida y sencilla en que podemos incluir ambas en el programa y cambiar de un modo a otro es escribiendo ambas y comentar/descomentar la que queramos utilizar

Funciones

Una función necesaria para que funcione correctamente la función stepper es la función SetDirection()

Esta función es la que lee y modifica las variables globales Direction y Steps que al fin y al cabo son las que rigen el movimiento de nuestros motores.

Esta es la función:

```
void SetDirectionH()
{
    if(DirectionH)
        StepsH++;
    else
        StepsH--;

    StepsH = ( StepsH + 8 ) % 8 ;
}
```

*Nota

Usaremos el código del motor horizontal, pero es exactamente igual al vertical, cambiando las H por V y poniendo los pines correspondientes en las funciones digitalWrite

Como hemos dicho, el sentido de giro viene dado por la variable Direction.

En el motor real, el sentido lo controlamos modificando el orden en que excitamos las bobinas. Por eso, en función de la variable Direction, incrementaremos o disminuirémos el paso en el que estamos.

2. Anejos a la memoria

Por ejemplo:

Esto quiere decir que, mientras Direction valga 1, el valor de Steps irá aumentando, lo cual se traducirá en un giro del motor.

Cuando Direction valga 0, el valor de Steps irá disminuyendo, lo cual se traducirá en un giro del motor, en sentido contrario al anterior.

Ahora tenemos el problema de que sólo tenemos 8 pasos posibles, por tanto, no podemos dar más de 8 pasos en una dirección, porque al estar en el paso 8 y tratar de situarnos en el paso 9 será imposible. ya que no existe el paso 9.

En realidad en paso 9 es el paso 1. Esto quiere decir que necesitamos una forma de que cuando llegemos al paso 8 volvamos a pasar al 1.

Esto lo conseguimos con la última línea del programa, que lo que hace es devolver el resto de la operación $(Steps + 8)/8$ lo cual daría 1 para Steps = 9

Con todo, podemos ver cómo al acabar de ejecutarse, la función SetDirection() habrá asignado a Steps el valor del próximo paso al que debe ser el movido el motor.

Nota*

Esto era un ejemplo. Puesto que en las matrices, el primer elemento es el [0][0], los pasos en lugar de ir del 1 al 8 van del 0 al 7.

Aunque el código sigue siendo el mismo, ya que en este realidad el paso 8 se correspondería con el paso 0 y el resto de $(8 + 8)/8$ es 0.

Además, nótese de que sumamos 8 a Steps porque cuando Direction sea 0, y estemos en el Step 0, si seguimos disminuyendo obtendremos el paso -1. Al sumarle 8, el resto de $(-1 + 8)/8$ es 7 y la cuenta puede seguir disminuyendo.

Hasta aquí los elementos que deberemos definir. Ahora vamos a entrar con la función propiamente dicha.

```
void stepperH()
{
    digitalWrite( IN1, Paso[StepsH][ 0] );
    digitalWrite( IN2, Paso[StepsH][ 1] );
    digitalWrite( IN3, Paso[StepsH][ 2] );
    digitalWrite( IN4, Paso[StepsH][ 3] );

    SetDirectionH();
}
```

Cuando la función Stepeper() es llamada, el motor es movido al paso número Step. Una vez hecho esto la función SetDirection() asigna a Step el valor del próximo paso, en función de la variable Direction, cómo ya hemos visto.

2. Anejos a la memoria

Una vez hayamos llamado a la función el número correspondiente de veces, antes de pasar a la siguiente instrucción escribiremos un valor bajo de todos los pines del motor, para que el motor no esté consumiendo energía. Esto podemos hacerlo ya que tenemos sistemas de autorretención en todos las transmisiones.

2.7.6. Funciones de control de los steppers

Estas funciones son las que posibilitan el control de los motores en función de los parámetros solares

Se han realizado dos funciones, una para el movimiento de elevación y otra para el de rotación, que en función del incremento de posición a realizar nos dan el número de pasos que debemos mover a los motores

Movimiento de elevación:

Necesitamos las variables globales:

barr_est: longitud de la barra estática, en mm

barr_movil: longitud de la barra móvil, en mm

tect: inclinación de la barra estática respecto el mástil vertical, en grados

paso_sf es el paso del sinfín, en mm

ir_elev: es la reducción entre la salida del stepper y el tornillo

g_elev: es el paso angular del stepper de elevación calculado como $^{\circ}\text{girados/paso}$

Su declaración:

```
int barr_est, barr_movil, tect;  
double g_elev, ir_elev, paso_sf;
```


2. Anejos a la memoria

Y el código de la función:

```
int elevacion(double A, double B){ //siendo B>A ya que a
mayor ángulo, menor x
    double steps, D1, D2, x1, x2, i_elev;
    //D son variables auxiliares para calcular el ángulo de
la formula
    //x
    int result;
    D1= 180-A-tect;
    D2= 180-B-tect;
    x1=((2*barr_est*cosd(D1))+sqrt(sq(2*barr_est*cosd(D1))-
4*(sq(barr_est)-sq(barr_movil))))/2;
    x2=((2*barr_est*cosd(D2))+sqrt(sq(2*barr_est*cosd(D2))-
4*(sq(barr_est)-sq(barr_movil))))/2;

    i_elev = ir_elev * paso_sf;
    steps=(360*(x1-x2))/g_elev;
    steps = steps*(1/i_elev);
//La operación original es: (x1-x2)/(g*ir_elev*(p/360))

//que ha sido reorganizada en dos pasos para evitar
desbordamientos
    result= (int) steps;

if ((steps-result) > 0.5) {
    result++;
}
return result;
}
```

A la función hay que introducirle los dos valores entre los que nos queremos mover, y nos devolverá los pasos necesarios para llevarlo a cabo.

El proceso de cálculo es el explicado en en el apartado **2.4 Análisis del mecanismo de elevación**

Finalmente, si el movimiento girado no es múltiplo de la resolución, obtendremos un numero de pasos decimal, que redondearemos al valor mas próximo.

El sentido de giro es controlado por medio de una variable global

2. Anejos a la memoria

Movimiento de rotación:

Necesitaremos las variables globales:

i_rot: relación de transmisión entre la salida del motor y el seguidor

g_rot: paso angular del motor de rotación

Las declaramos como:

```
double i_rot, g_rot;
```

Y el código de la función:

```
int rotacion(double A, double B){
    double steps;
    int result;
    steps = (B-A)/g_rot;
    steps = steps * (1/i_rot);
    result= (int) steps;

    if ((steps-result) > 0.5) {
        result++;
    }
    return result;
}
```

A la función hay que introducirle los dos valores entre los que nos queremos mover, y nos devolverá los pasos necesarios para llevarlo a cabo.

El proceso de cálculo es el mismo explicado en el apartado anterior y una vez mas, redondeamos el número de pasos del motor en caso de tener un movimiento que no sea múltiplo de la resolución

Paro de los motores

Motor de rotación:

```
void pararH()
{
    digitalWrite( IN1, 0 );
    digitalWrite( IN2, 0 );
    digitalWrite( IN3, 0 );
    digitalWrite( IN4, 0 );
}
```

Motor de elevación.

```
void pararV()
{
    digitalWrite( IN5, 0 );
    digitalWrite( IN6, 0 );
    digitalWrite( IN7, 0 );
    digitalWrite( IN8, 0 );

}
```

2.7.7. Control del tiempo

El tipo de control que hemos implementado presenta un problema. En caso de que haya un corte en la alimentación, la fecha es reinicia a sus valores por defecto, que en Arduino es el 1 de Enero de 1970

Para evitar esto, utilizaremos un reloj de tiempo real RTC (Real Time Clock) que consiste en un circuito integrado I2C con un microchip DS1307 que incluye un pequeño reloj con una batería (*una pila de botón*) de respaldo.

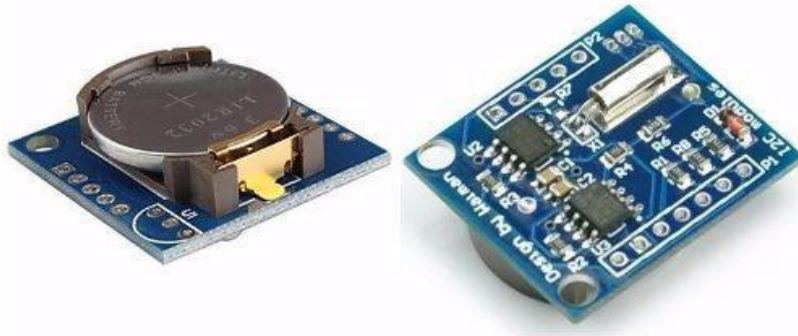


Fig. 41 Ejemplo de chip comercial pre montado

Para poder implementarlo únicamente deberemos cargar las librerías necesarias para su control y ponerlo en hora.

Las librerías que debemos incluir son:

```
#include <Wire.h>
#include <DS1307RTC.h>
```

La librería Wire.h es la que permite comunicar con dispositivos I2C

La librería DS1307RTC.h es la que gestiona el microchip DS1307 para la librería Time.

Para sincronizar el RTC con el Arduino deberemos incluir en el bucle *setup()* la línea:

```
setSyncProvider(RTC.get)
```

Finalmente, para ponerlo en hora solo tenemos que incluir en este mismo bucle la función *setTime* como hemos visto en el apartado 2.6.3.

2.7.8. Programa completo

El programa final, listo para implementar en Arduino es:

```
#include <TimeLib.h>
#include <Wire.h>
#include <DS1307RTC.h>

//Constantes geográficas
// phi: latitud
const float phi =39.99;

// L: longitud del punto
const double L = -0.068971;

// Lm: longitud del huso horario del punto instalación
const int Lm = 0;

// delta: declinacion
double delta ;
//auxiliares para la division:
long dx;
double dy,dz;

// h: angulo horario (salida,s/puesta,p)
// h= 15*(12-TSV)
double h, hp, hs;

// TSV: hora solar verdadera (salida,s/puesta,p)
// TSV= HO + ep + ET + (Lm - L)/15
double TSV,TSVs,TSVp;
// HO: hora oficial (formato por determinar)
//-función HO()-
// hora oficial de puesta/salida del Sol
double HOp,HOs, HOs_next;

// ET: Ecuacion del tiempo
// ET= 8.87*sin(2B)-7.53*cos(2B)-1.5*sin(B)
double ET; //se puede estudiar hacerla float
double B;

// altitud: altitud solar
// als=sin(phi)*sin(delta)+cos(phi)*cos(delta)*cos(h)

// ángulo acimutal
// (cos(delta)*sin(h))/cos(als)
// azss: acimut salida del sol
```

2. Anejos a la memoria

```
double azss, azss_next;

double al_prev, al_now, az_prev, az_now;
//al_now: altitud en el instante de calculo actual
//al_prev: altitud en el instante de calculo anterior
//az_xx: lo mismo pero con el acimut

const double al_fc, az_fc;
//al_fc: altitud Solar cuando las placas estan totalmente
elevadas
//az_fc: acimutal del FC_H

// z: día del año (1-365)
// -función z()- necesita las variables:
bool flag=0; //inicializar a 1 si la fecha de inicio es a
partir del 29 de febrero de un año bisiesto
int fin_bisiesto; //si el año de inicializacion es
bisiesto, asignarle el valor del año siguiente

//OTRAS_____
_____

//variable control día/noche
bool fin_dia;

//variable control día siguiente
int dia_actual;

//variables control a lo largo del día
bool crepusculo, mediodía;

//CONTROL
STEPPERS_____
_____

//Para los pines de control del stepper Horizontal:
#define IN1 12
#define IN2 11
#define IN3 10
#define IN4 9

//Para los pines de control del stepper Vertical:
#define IN5 8
#define IN6 7
#define IN7 6
#define IN8 5

//Para los finales de carrera
#define FC_V 4
#define FC_H 3
```

2. Anejos a la memoria

```
//variables

int StepsH,StepsV;
int n_al, n_az;
//n_al: numero de pasos que debe dar el motor de elevacion
para moverse a la proxima posición
//n_az: numero de pasos que debe dar el motor de rotacion
para moverse a la proxima posición

const int t_V, t_H;
//t_V, t_H: tiempo de espera enter pasos del motor, define
la velocidad de giro del motor

bool DirectionH, DirectionV;

//Control medio paso

int Paso [ 8 ][ 4 ] =
{
    {1, 0, 0, 0},
    {1, 1, 0, 0},
    {0, 1, 0, 0},
    {0, 1, 1, 0},
    {0, 0, 1, 0},
    {0, 0, 1, 1},
    {0, 0, 0, 1},
    {1, 0, 0, 1}
};

//Control Paso completo

//int Paso [ 4 ][ 4 ] =
// {    {1, 1, 0, 0},
//      {0, 1, 1, 0},
//      {0, 0, 1, 1},
//      {1, 0, 0, 1}
// };

//Para cambiar entre un modo u otro comentar o descomentar
el tipo deseado

//CONSTANTES           DE           LA           TRANSMISION           Y
DINÁMICA
_____

//Elevación
```

2. Anejos a la memoria

```
const int barr_est, barr_movil, tect;
//barr_est longitud de la barra estatica
//barr_movil longitud de la barra movil
//tect inclinación de la barra estatica

const double g_elev, ir_elev, paso_sf;
//paso_sf es el paso del sinfin, en mm
//ir_elev es la reduccion entre la salida del stepper y el
tornillo EN FORMATO DECIMAL
//g_elev es el paso angular del stepper de elevación
(°/paso)

//Rotacion
const double i_rot, g_rot;
//i_rot es la reduccion en el motor de rotación EN FORMATO
DECIMAL (será necesaria de unos 1/200)
//g_rot paso angular motor de rotacion (°/paso)

void setup() {

    //Modo pines steppers
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(IN3, OUTPUT);
    pinMode(IN4, OUTPUT);
    pinMode(IN5, OUTPUT);
    pinMode(IN6, OUTPUT);
    pinMode(IN7, OUTPUT);
    pinMode(IN8, OUTPUT);
    pinMode(FC_V, INPUT);
    pinMode(FC_H, INPUT);

    //Sincronizar RTC
    setSyncProvider(RTC.get)

}

void loop() {

while (fin_dia ==0)
{
    // rutina principal del día

    time_t inst_dia;
    inst_dia = now(); //se le puede sumar un tiempo de
movimiento aproximado, para que llegue a la posicion en la
que deberia estar
    //en el instante que deje de moverse
```

2. Anejos a la memoria

```
//elevación_____

    if ( (als( HO(inst_dia) , z(inst_dia) ) > al_fc) &&
crepusculo == 0 ) //Espera a que el Sol se eleve lo
suficiente
    {
        crepusculo = 1; //se inicializara a 0 al final
del día
    }

    while(crepusculo==1) //cuando el Sol esta a altitud
suficiente, empezara a mover las placas
    //mientras el Sol no este a altitud suficiente, este
bucle es omitido y únicamente se lleva a cabo el movimiento
de rotacion
    {
        al_now = als( HO(inst_dia) , z(inst_dia) );

        if ( (al_now < al_prev) && (mediodía == 0) )
//alcanzado el mediodía, el Sol empieza a descender
//por tanto, la altitud solar actual es menor a la
anterior
        {
            mediodía = 1; //evitamos que vuelva a entrar
al bucle
            DirectionV = 0; //invertimos la direccion de
giro del motor, para que empiece a ascender las placas
        }

        n_al = elevacion (al_prev, al_now); //calculamos los
pasos del motor necesarios para mover a la nueva posición
        al_prev=al_now; //dejamos al_prev listo para el
proximo calculo (al inicio el día, al_prev será conocido,
por estar la placa elevada hata el final de carrera)

        while (n_al>0) //bucle que hace girar el motor
n_al pasos, la dirección ha sido anteriormente definida
según el momento del día
        {
            if (digitalRead(FC_V)==0) //para que el
motor no siga girando si se alcanza la elevación máxima
            {
                stepperV();
                n_al--;
                delay(t_V);
            }
        }
        pararV(); //cesamos la señal en los pines del
stepper
    }
}
```


2. Anejos a la memoria

```
    }

    //rotacion_____ (no tiene relacion
con la elevacion, salvo la variable inst_dia, para tener
posiciones consistentes)

    az_now = azs( HO(inst_dia) , z(inst_dia) );

    n_az = rotacion(az_now, az_prev);
    az_prev = az_now;

    while (n_az>0) //bucle que hace girar el motor n_az
pasos, durante el día DirectionH = 1
    {
        stepperH();
        n_az--;
        delay(t_H);
    }

    pararH(); //paramos el stepper

    delay(300000); //delay que controla tiempo entre
calculos, en milisegundos (300000 = 5 minutos)

    if (HO(now())>HOp) //acaba el día, este bucle se
ejecuta una vez al final del día
    {
        fin_dia=1; //marca la salida del bucle del día

        //obtenemos dia_actual
        dia_actual = z(now()+86400); //para añadir un día
le sumamos los segundos de un día
        //recalculamos HOP y HOs para el día siguiente
        dx = 360L*(284+dia_actual);
        dy = 365;
        dz=dx/dy;

        delta = 23.45*sind(dz);

        hp = acosd((-sind(phi)*sind(delta))/(cosd(phi)*cosd(delta))); //h
puesta

        hs = -hp; //h salida

        TSVp = 12+(hp/15);
        TSVs = 12+(hs/15);

        B = (360/364) * (dia_actual-81);
```

2. Anejos a la memoria

```
ET = 8.87*sind(2*B) - 7.53 * cosd(2*B) -
1.5*sind(B);

HOp = TSVp-(ET/60)+ (L-Lm)/15; //puesta Sol
HOs = TSVs-(ET/60)+ (L-Lm)/15; //salida Sol
//recordar en ambos ep!!! que al final no se
pone

//posición salida Sol
azss= acosd(cosd(delta)*sind(hp));

//posición inmediatamente posterior a la
salida Sol

HOs_next = HOs + 0.01; //Hos esta en formato
decimal
azss_next = azs(HOs_next,dia_actual);

//mover a la posición de salida del Sol

//elevacion_____

while(digitalRead(FC_V)==0)//si no esta la
placa elevada
{
    stepperV(); //seguimos elevando las placas
    delay(t_V); //a la velocidad seleccionada
}

pararV();

//ya tenemos la placa en su correcta inclinación

//rotacion

DirectionH = 0; //Durante el día, vale 1;
puesto que ahora moveremos el seguidor en sentido contrario
para volver a la posición
//de salida, invertimos el giro

while (digitalRead(FC_H)==0)
{
    stepperH();
    delay(t_H);
}
//ahora el seguidor está mirando hacia el este
```

2. Anejos a la memoria

```
        n_az = rotacion(az_fc,azss);//recomendable
poner az_fc a 0° respecto el este

        if(azss<azss_next)//si el Sol sale por el Sud-
Este
        {
            DirectionH = 1; //ya que el incremento
hasta la posicion aszz es en sentido Solar
            //en caso de que azss_next>azss deberemos
seguir girando en el mismo sentido en que
            //girabamos para llegar al FC, por tanto
no modificamos el sentido
        }

        while (n_az<0) //girará el motor en la
direccion que queremos
        {
            stepperH();
            n_az--;
            delay(t_H);
        }

        pararH();

        //ya tenemos el seguidor encarado hacia azss
    }
}

if (fin_dia ==1)
{
    if (z(now())==dia_actual && HO(now())>HOs) //esperamos a
que salga el sol el día siguiente
    {
        fin_dia=0;
        crepusculo = 0;
        mediodía = 0;
        DirectionH= 1; //para que pase lo que pase, eal inicio
del día el seguidor se mueva en el sentido correcto
        DirectionV = 1; //puesto que las placas ya estan
totalmente elevadas, cuando volvamos a entrar al bucle del
día
            // las placas empezaran a descender. cabe
recordar que las palaas ascenderan cuando se active el
bucle mediodía

            az_prev = azss;
            al_prev = al_fc;
        }
    }
}
```

2. Anejos a la memoria

```
}

//                                DEFINICIÓN                                DE
FUNCIONES_____
__#####

//
TIEMPO_____

//Hora actual, HO()

double HO(time_t t )
{

    double result,horas,mints,segs;

    horas = hour(t);
    mints = minute(t);
    segs = second(t);

    result = horas + mints/60 + segs/3600;
    return result;

}

//Día en formato 0 - 366
int z(time_t t)
{

    int
days[13]={0,0,31,59,90,120,151,181,212,243,273,304,334};

    int result = day(t) + days[month(t)];

    //Gestión de los años bisiestos:

    if (day(t)==29 && month(t)==2)
    {
        flag = 1;
        fin_bisiesto = year(t)+1;
    }

    if (fin_bisiesto == year(t))
    {
        flag = 0;
    }
}
```

2. Anejos a la memoria

```
        fin_bisiesto = 1900;
    }

    if (flag == 1 && month(t)>2)
    {
        result = result + 1;
    }

    return result;
}

//
TRIGONOMETRIA

//seno en grados de un ángulo
double sind(double x){
    double result;
    const double pi = 3.1415926535;
    result = sin(x*pi/(180));
    return result;
}

//coseno en grados de un ángulo
double cosd(double x){
    double result;
    const double pi = 3.1415926535;
    result = cos(x*pi/180);
    return result;
}

//Arco-seno en grados de un angulo
double asind(double x){

    const double pi = 3.1415926535;
    double result;

    result = asin(x)*180/pi;
    return result;
}

//Arco-coseno en grados de un angulo
double acosd(double x){

    const double pi = 3.1415926535;
    double result;

    result = acos(x)*180/pi;
```

2. Anejos a la memoria

```
    return result;
}

// _____ POSICIÓN SOLAR

//Altitud Solar, en grados
double als(double hora, int día){

    double result;

    dx = 360L*(284+día);
    dy = 365;
    dz=dx/dy;

    delta = 23.45*sind(dz);

    B = (360/364) * (día-81);

    ET = 8.87*sind(2*B) - 7.53 * cosd(2*B) - 1.5*sind(B);

    //TSV = HO() +ET/60 + (Lm-L)/15;

    TSV = hora +ET/60 + (Lm-L)/15;
    h= 15*(12-TSV); //ángulo horario

    result=asind(sind(phi)*sind(delta)+cosd(phi)*cosd(delta)*
    cosd(h));

    return result;
}

//Azimut Solar en grados
double azs (double hora, int día){
    double result;
    double altitud;
    altitud = als(hora, día);
    result= acosd((cosd(delta)*sind(h))/cosd(altitud));
    return result;
}

// _____ STEPPERS

void stepperH()
{
    digitalWrite( IN1, Paso[StepsH][ 0] );
    digitalWrite( IN2, Paso[StepsH][ 1] );
}
```

2. Anejos a la memoria

```
        digitalWrite( IN3, Paso[StepsH][ 2] );
        digitalWrite( IN4, Paso[StepsH][ 3] );

        SetDirectionH();
    }

void SetDirectionH()
{
    if(DirectionH)
        StepsH++;
    else
        StepsH--;

    StepsH = ( StepsH + 8 ) % 8 ; //Modo control medios
pasos
    // StepsH = ( StepsH + 4 ) % 4 ; //Modo control pasos
completos
}

void stepperV()
{
    digitalWrite( IN1, Paso[StepsV][ 0] );
    digitalWrite( IN2, Paso[StepsV][ 1] );
    digitalWrite( IN3, Paso[StepsV][ 2] );
    digitalWrite( IN4, Paso[StepsV][ 3] );

    SetDirectionV();
}

void SetDirectionV()
{
    if(DirectionV)
        StepsV++;
    else
        StepsV--;

    StepsV = ( StepsV + 8 ) % 8 ; //Modo control medios
pasos
    //StepsV = ( StepsV + 4 ) % 4 ; //Modo control pasos
completos

}

int elevacion(double A, double B){ //siendo B>A ya que a
mayor ángulo, menor x
    double steps, D1, D2, x1, x2, i_elev;
    //D son variables auxiliares para calcular el ángulo de
la formula
    //x
```

2. Anejos a la memoria

```
int result;
D1= 180-A-tect;
D2= 180-B-tect;
x1=((2*barr_est*cosd(D1))+sqrt(sq(2*barr_est*cosd(D1))-
4*(sq(barr_est)-sq(barr_movil))))/2;
x2=((2*barr_est*cosd(D2))+sqrt(sq(2*barr_est*cosd(D2))-
4*(sq(barr_est)-sq(barr_movil))))/2;

i_elev = ir_elev * paso_sf; //i_elev es la reduccion
total de la transmisión en mm/vuelta del stepper
steps=(360*(x1-x2))/g_elev; //La operación original es:
(x1-x2)/(g*ir_elev*(p/360))
steps = steps*(1/i_elev); //que ha sido reorganizada
en dos pasos para evitar desbordamientos
result= (int) steps;

    if ((steps-result) > 0.5) {
        result++;
    }
return result;
}

int rotacion(double A,double B){
double steps;
int result;
steps = (B-A)/g_rot;
steps = steps * (1/i_rot);
result= (int) steps;

    if ((steps-result) > 0.5) {
        result++;
    }
return result;
}

//Parar steppers

void pararH()
{
    digitalWrite( IN1, 0 );
    digitalWrite( IN2, 0 );
    digitalWrite( IN3, 0 );
    digitalWrite( IN4, 0 );

}

void pararV()
{
    digitalWrite( IN5, 0 );
```


2. Anejos a la memoria

```
digitalWrite( IN6, 0 );  
digitalWrite( IN7, 0 );  
digitalWrite( IN8, 0 );
```

```
}
```

2.7.9. Inicialización

En primer lugar debemos poner en hora el reloj, usando la función *setTime* es importante hacer esto en el bucle *setup* porque de hacerlo junto el programa principal el tiempo estaría redefiniéndose constantemente y la rutina no funcionaría.

Constantes

Las constantes deben ser inicializadas de acuerdo a lo dispuesto en el apartado

2.7.1 Constantes necesarias

Variables

Las variables que debemos inicializar, son:

Variable	Valor	Situación
flag	1	A partir del 1 de Marzo de un año bisiesto
	0	Cualquier otro momento
fin_bisiesto	<i>valor del año próximo</i>	Año actual bisiesto
	<i>no inicializar</i>	Año actual no-bisiesto
fin_dia	1	Se ha puesto el Sol
	0	Es de día
crepusculo	1	El Sol ha salido completamente
	0	Aún no ha amanecido
mediodía	1	El Sol está descendiendo
	0	El Sol está ascendiendo
dia_actual	<i>día de hoy (1-366) +1</i>	Siempre

Cada variable deberá ser inicializada con el valor correspondiente en función de la condición de la columna "Situación" que se cumpla el momento de la instalación.

En el caso de que la instalación se lleve a cabo durante el día, se debe poner el seguidor en la posición aproximada que debería estar, sin más problema porque al final de cada día el sistema se auto-ajusta.

Si la instalación se realiza una vez puesto el Sol, se deberá situar el seguidor con las placas a máxima inclinación y el acimutal correspondiente. Igual que en caso anterior, el error cometido se arrastrará hasta el día siguiente. En este caso los valores de salida y puesta del sol (HOs, HOp) también deberán ser inicializados

2.7.10. Conexionado

La mayoría de conexiones son bastante sencillas.

En lo que en el programa hemos llamado final de carrera, en realidad será un pulsador. Para minimizar el consumo de energía los utilizaremos con una resistencia en modo pull down, cuyo esquema eléctrico es:

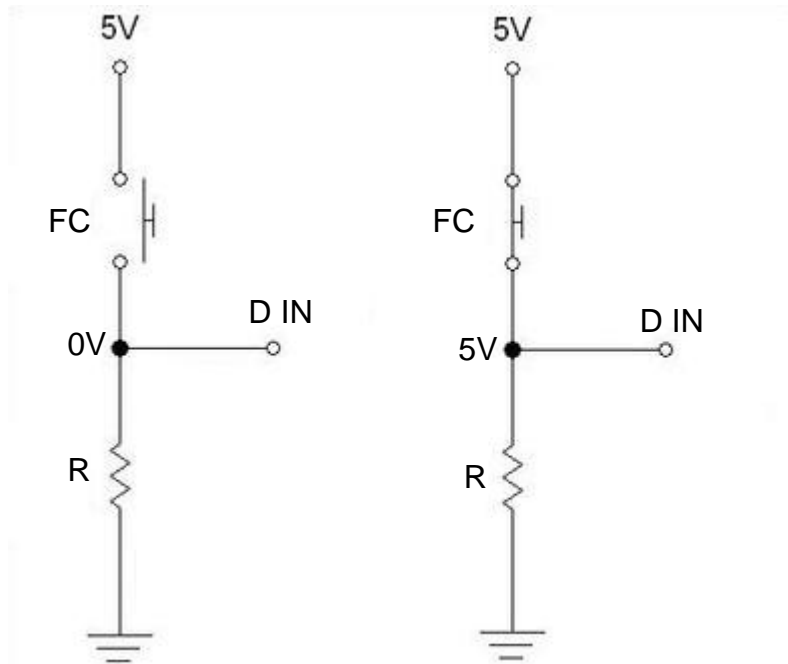


Fig. 42

Donde FC es el final de carrera, R una resistencia (típicamente de 10k) y D IN la entrada digital de Arduino

Para el control de los steppers utilizaremos dos circuitos integrados L293D. El esquema de conexión es un poco más complejo, pero está en el datasheet del fabricante en el apartado **3. Planos**

A la hora de materializar el proyecto necesitaremos mantener la parte electrónica alejada de las inclemencias del tiempo. Por lo que necesitaremos introducir el sistema en un entorno con cierta estanqueidad. La forma más económica de hacer esto es utilizar una caja de empalmes comercial que sea apta para exteriores. De este modo podremos mantener todos los componentes electrónicos en su interior. Por motivos prácticos, la caja estará situada sobre la barra que apoya en el mástil.

A los cables que traen por defecto los steppers se les añadirá el cable necesario para su conexión con el microprocesador. Estas uniones se realizarán con regletas de conexión.

2.8. Selección de materiales estructurales

Necesitamos elegir el material del cual se construirá el seguidor, puesto que en apartados posteriores necesitaremos hacer cálculos relacionados con las características físicas de la estructura. Como su peso propio, dimensionamiento de los anclajes, selección de los materiales para las uniones, etc.

Para tomar la decisión, llevaremos a cabo el método de selección multicriterio aplicado a los materiales contemplados:

Madera

Es barata, fácil mecanizado (corte, taladro, uniones móviles), y ligera (550 – 820 Kg/m³)

Sin riesgo de oxidación galvánica. En cambio, se degrada en presencia de humedad (aunque hay tratamientos para ralentizar este proceso), no se puede soldar (debe ser pegada), tiene poca resistencia, y se deforma permanentemente con facilidad.

Tubos PVC

Es barato, fácil de mecanizar (corte, taladro, uniones y juntas comerciales), es muy ligero y tiene una resistencia a la intemperie muy elevada.

Sin riesgo de oxidación galvánica.

No obstante, es frágil y soporta pocos esfuerzos.

Perfiles Aluminio

Relativamente ligero, fácilmente mecanizable, químicamente estable (soporta bien la oxidación galvánica y la intemperie). resistencia medía.

Precio bastante elevado

Hierro

Precio asequible, fácilmente mecanizable, soldadura fácil, buen precio.

Pesado (7478kg/m³), riesgo de oxidación galvánica, poca resistencia a la intemperie. Pesado

Acero Galvanizado y Acero Inoxidable

Buenas características mecánicas. Mecanizado y soldadura aceptables. Más barato que el Aluminio.

Peso elevado

De entrada, vamos a eliminar el PVC y la madera de la comparativa. El PVC porque no tiene las características estructurales necesarias para soportar las cargas que aparecerán. Y la madera porque a raíz de estar expuesta continuamente a la intemperie se degradará rápidamente.

2. Anejos a la memoria

Núm.	Material	ρ (Kg/m ³)	Precio	E (Pa)	Mecanizado	Oxidación galvánica	Resistente a la Intemperie
1	Perfil Aluminio	2698,4	Alto	$70 \cdot 10^9$	Fácil	Baja	Elevada
2	Hierro	7874	Bajo	$190 \cdot 10^9$	fácil-Medio	Medía	Medía-Baja (necesita tratamiento)
3	Acero Inox	7960	Alto	$210 \cdot 10^9$	Medio-Alto	Baja (depende del % Cr)	Elevada
4	Acero Galvanizado	7850	Medio	$210 \cdot 10^9$	Medio	Nula	Medía-Elevada

En primer lugar, establecemos una preferencia entre atributos:

1º- Módulo de Elasticidad (E)

Ante todo, queremos que el seguidor sea resistente.

2º- Precio

El objetivo del proyecto es hacer un seguidor lo más barato posible.

3º- Resistencia a la intemperie

Puesto que el seguidor estará todo el tiempo bajo esta condición.

4º- Oxidación galvánica

Esto puede hacer que las uniones se debiliten con el tiempo. Si queremos que el seguidor sea duradero debemos evitarla.

5º- Densidad

Cuanto más baja mejor, pero tampoco pasa nada si el seguidor es muy pesado, ya que esto favorecerá la resistencia a las ráfagas de viento.

6º- Mecanizado

Finalmente, intentaremos que el seguidor sea lo más fácil de construir posible. Esta característica ha sido situada la última debido a que en proporción a todo el tiempo que esperamos que el seguidor está operativo, los costes de mecanizado pueden estar justificados si todos los demás requisitos son satisfechos.

2. Anejos a la memoria

A continuación, establecemos el peso de cada atributo, por medio de la comparación con otros atributos.

	E	Precio	Intemperie	Oxidación	ρ	Mecanizado
E	1	3	5	6	7	9
Precio	1/3	1	3	4	6	8
Intemperie	1/5	1/3	1	2	4	7
Oxidación	1/6	1/4	1/2	1	4	6
ρ	1/7	1/6	1/4	1/4	1	2
Mecanizado	1/9	1/8	1/7	1/6	1/2	1

Y calculamos su peso total como el producto de todas las puntuaciones elevado a 1/(el número de atributos)

E	$(1*3*5*6*7*9)^{1/6}$	= 4,22276756
Precio	$(1/3*1*3*4*6*8)^{1/6}$	= 2,40187391
Intemperie	$(1/5*1/3*1*2*4*7)^{1/6}$	= 1,24551643
Oxidación	$(1/6*1/4*1/2*1*4*6)^{1/6}$	= 0,89089872
ρ	$(1/7*1/6*1/4*1/4*1*2)^{1/6}$	= 0,3792656
Mecanizado	$(1/9*1/8*1/7*1/6*1/2*1)^{1/6}$	= 0,23427793
	Sumatorio	= 9,37460014

A continuación, obtenemos el peso en tanto por uno.

$$peso_{t.p.u.i} = \frac{peso_i}{\sum peso_i}$$

E	0,4504
Precio	0,2562
Intemperie	0,1329
Oxidación	0,0950
ρ	0,0405
Mecanizado	0,0250
	Sumatorio = 1

Ahora, elaboramos la matriz de decisión, dónde se relaciona cada material con los atributos que posee. Puntuados entre 0 y 1 con incrementos iguales e incluyendo siempre ambos extremos. Si dos materiales están empatados tienen la misma puntuación, pero los intervalos deben seguir siendo iguales.

Material	E	Precio	Intemperie	Oxidación	ρ	Mecanizado
Perfil Aluminio	0	0	1	0,5	1	1
Hierro	0,5	1	0	0	0	0,66
Acero Inoxidable	1	0	1	0,5	0,33	0
Acero Galvanizado	1	0,5	0,5	1	0,66	0,33

2. Anejos a la memoria

Finalmente, solo queda multiplicar la puntuación de cada característica por el peso de dicha característica. Y todos los productos para cada material.

Esto lo haremos de forma más rápida con una multiplicación de matrices en Excel. Para ello utilizaremos la matriz de decisión que acabamos de obtener y los valores de los pesos en t.p.u. de cada atributo ordenados en una matriz columna.

Material	E	Precio	Intempérie	Oxidación	ρ	Mecanizado	Matriz Pesos t.p.u.	Nota final
Perfil Aluminio	0,00	0,00	1,00	0,50	1,00	1,00	0,45	0,25
Hierro	0,50	1,00	0,00	0,00	0,00	0,66	0,26	0,50
Acero Inoxidable	1,00	0,00	1,00	0,50	0,33	0,00	0,13	0,64
Acero Galvanizado	1,00	0,50	0,50	1,00	0,66	0,33	0,10	0,77
							0,04	
							0,02	

Con estos resultados, ordenamos las opciones según la nota obtenida:

- 1º- Acero Galvanizado
- 2º- Acero Inoxidable
- 3º- Hierro
- 4º- Perfil Aluminio

Por tanto, el material utilizado para construir el seguidor será Acero Galvanizado.

2.9. Cálculos estructurales.

2.9.1. Planteamiento

A la hora de construir un seguidor, las variables más significativas a tener en cuenta a la hora de hacer cálculos estructurales son el peso propio y la acción del viento.

Vamos a realizar una pequeña simplificación del proceso de cálculo de la estructura a modo ilustrativo.

Para ello tomamos como referencia un esquema simplificado del seguidor:

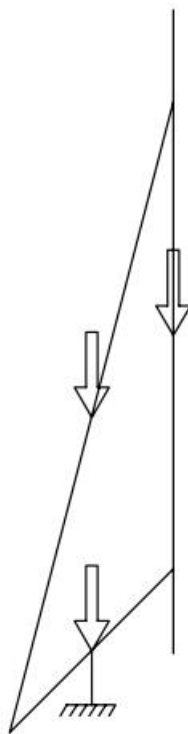


Fig. 43

Las flechas que apuntan hacia abajo representan el peso propio de las barras. Cada carga adopta el valor del peso total de su barra. Excepto en caso de la barra vertical, en cuyo caso la carga deberá incluir tanto el peso propio de la barra como el de todos los elementos que formen parte de ella (placas solares, actuadores)

Además, cada carga está localizada en el centro de gravedad de cada barra.

2. Anejos a la memoria

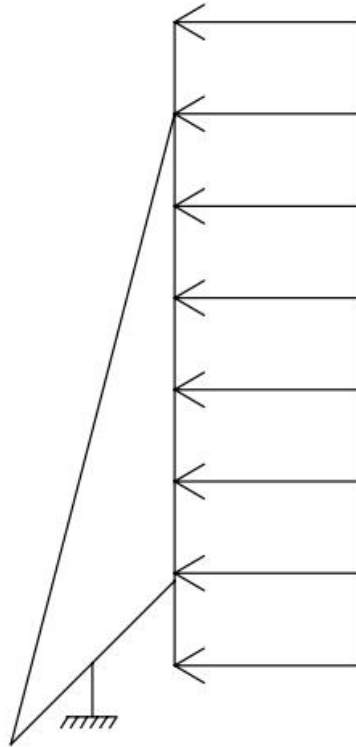


Fig. 44

La carga distribuida horizontalmente representa el viento, y afecta a toda la superficie vertical.

2. Anejos a la memoria

Para obtener el sistema final, debemos sumar ambos casos:

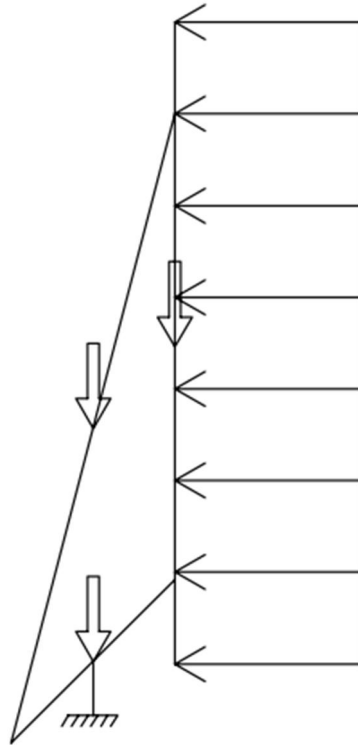


Fig. 45

Para obtener las reacciones de las barras y en el apoyo, necesitamos conocer el valor de las cargas puntuales.

El eso propio lo hemos simplificado como el peso total de al barra colocado en su centro de gravedad.

En este aspecto, la carga más complicada de obtener es la del viento.

A continuación vamos a realizar un ejemplo para la provincia de Castellón.

Para llevar a cabo el cálculo, seguiremos el Documento básico SE-AE, de Seguridad Estructural Acciones en la edificación.

Cabe decir que este documento está orientado a la edificación, y que el reglamento no contempla ni siquiera la instalación de placas solares sobre viviendas. Por tanto se adaptará el documento a nuestras necesidades.

2. Anejos a la memoria

Vamos al apartado 3.3 Viento / 3.3.2 Acción del viento, y dice así:

3.3.2 Acción del viento

1 La acción de viento, en general una fuerza perpendicular a la superficie de cada punto expuesto, o presión estática, q_e puede expresarse como:

$$q_e = q_b \cdot c_e \cdot c_p \quad (3.1)$$

siendo:

- q_b la presión dinámica del viento. De forma simplificada, como valor en cualquier punto del territorio español, puede adoptarse $0,5 \text{ kN/m}^2$. Pueden obtenerse valores más precisos mediante el anejo D, en función del emplazamiento geográfico de la obra.
- c_e el coeficiente de exposición, variable con la altura del punto considerado, en función del grado de aspereza del entorno donde se encuentra ubicada la construcción. Se determina de acuerdo con lo establecido en 3.3.3. En edificios urbanos de hasta 8 plantas puede tomarse un valor constante, independiente de la altura, de 2,0.
- c_p el coeficiente eólico o de presión, dependiente de la forma y orientación de la superficie respecto al viento, y en su caso, de la situación del punto respecto a los bordes de esa superficie; un valor negativo indica succión. Su valor se establece en 3.3.4 y 3.3.5.

Tal y como podemos ver, el valor de la carga se obtiene multiplicando tres variables:

$$q_e = q_b \cdot c_e \cdot c_p$$

q_b : vamos al Anejo D. Presión del viento / D.1 Presión dinámica para obtener un valor más exacto:

A partir del mapa, podemos observar que nos encontramos en una zona "A" de viento:

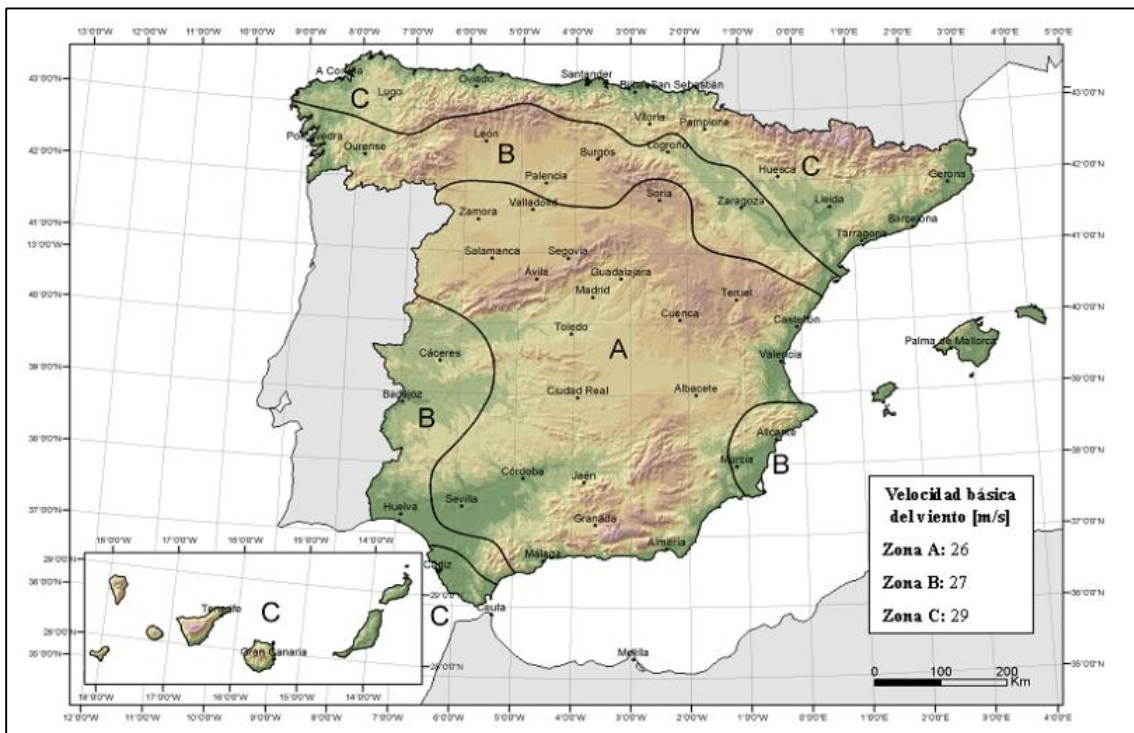


Fig. 46

2. Anejos a la memoria

A partir del documento es posible obtener un valor todavía más preciso en función de la densidad del aire, especialmente recomendado para instalaciones muy cercanas al mar. Como no es el caso nos limitaremos a toma uno de los valores recomendados en el punto 4:

- 4 El valor básico de la velocidad del viento en cada localidad puede obtenerse del mapa de la figura D.1. El de la presión dinámica es, respectivamente de $0,42 \text{ kN/m}^2$, $0,45 \text{ kN/m}^2$ y $0,52 \text{ kN/m}^2$ para las zonas A, B y C de dicho mapa.

Por tanto nuestro valor de q_b será:

$$q_b = 0,42 \text{ kN/m}^2$$

c_e : depende del entorno. El código nos remite al punto 3.3.3. donde se encuentra la siguiente tabla:

Grado de aspereza del entorno	Altura del punto considerado (m)							
	3	6	9	12	15	18	24	30
I Borde del mar o de un lago, con una superficie de agua en la dirección del viento de al menos 5 km de longitud	2,4	2,7	3,0	3,1	3,3	3,4	3,5	3,7
II Terreno rural llano sin obstáculos ni arbolado de importancia	2,1	2,5	2,7	2,9	3,0	3,1	3,3	3,5
III Zona rural accidentada o llana con algunos obstáculos aislados, como árboles o construcciones pequeñas	1,6	2,0	2,3	2,5	2,6	2,7	2,9	3,1
IV Zona urbana en general, industrial o forestal	1,3	1,4	1,7	1,9	2,1	2,2	2,4	2,6
V Centro de negocio de grandes ciudades, con profusión de edificios en altura	1,2	1,2	1,2	1,4	1,5	1,6	1,9	2,0

Supondremos la instalación en una villa de una planta sin conexión a red. Podemos suponer que estará más o menos aislada.

Por tanto podría ser zona III y con una altura del punto de instalación de hasta 6m

Con esos datos obtenemos:

$$c_e = 2,0$$

dependiendo colocación, cubierta, forma base, etc.

c_e : depende de la forma y orientación de la superficie a considerar. El reglamento nos remite a los puntos 3.3.4. Coeficiente eólico de edificios de pisos y al 3.3.5. Coeficiente eólico de naves y construcciones diáfanas.

2. Anejos a la memoria

Para nuestro caso, iremos a la tabla D.3 Paramentos verticales

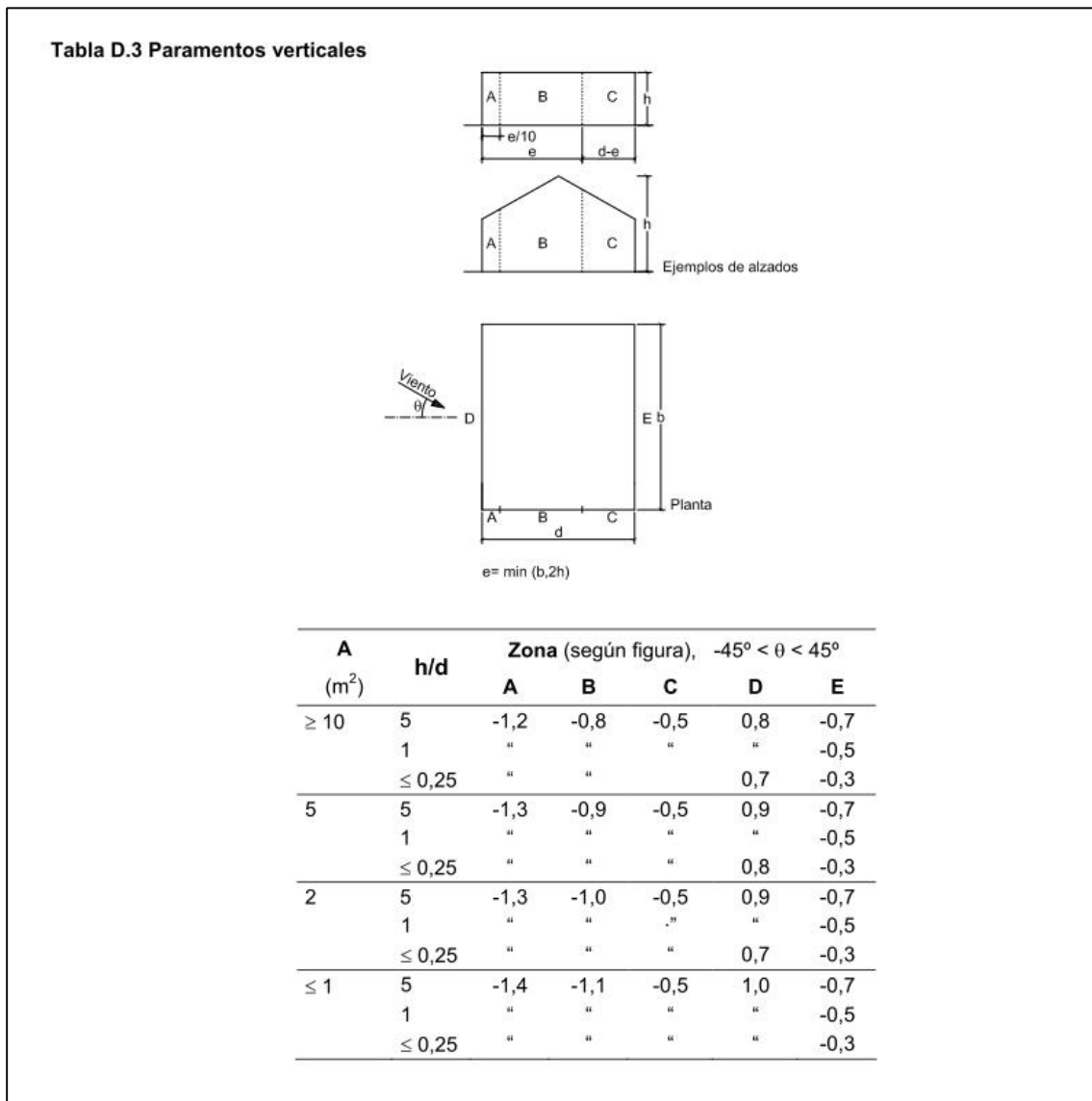


Fig. 47

Vamos a suponer el caso más desfavorable: cuatro placas de 1,6m x 1m situadas de forma vertical .

La cara dónde sopla el viento es la "D" y su área "A"

El espesor de las placas lo asociamos con "d"

El alto de las placas es "h"

El ancho de las placas es "b"

En nuestro caso:

$$A(\text{m}^2) = 1,6 \cdot 4 = 6,4\text{m}^2$$

$$d = 0,04\text{m}$$

$$h = 1,6 \cdot 2 = 3,2$$

$$b = 2\text{m}$$

2. Anejos a la memoria

Con ellas calculamos:

$$h/d = 80$$

*La columna de h/d va de 0,25 a 5, por tanto cogemos el valor máximo. Puesto que estos cálculos están pensados para el campo de la construcción no se contempla que existan elementos tan esbeltos.

$$e = \min(b, 2h) = \min(2 \cdot 6, 2) = 2\text{m}$$

$$e/10 = 0,2\text{m}$$

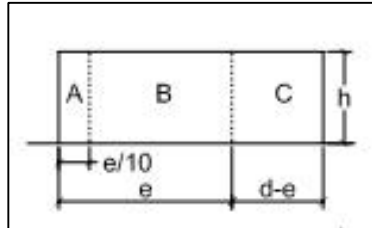


Fig. 48

Puesto que el espesor de la placa, “d” es menor que $e/10$ el canto de la placa será solo zona de influencia “A”. La placa es demasiado estrecha como para que hayan tres zonas de acción del viento.

Obtendremos el parámetro correspondiente a esta zona. Nos será útil para calcular los anclajes en próximos apartados.

Como nuestra área se encuentra entre 1 y 10 m² tenemos que interpolar para obtener los coeficientes, tal y como nos dice el reglamento.

$$C_{pe,A} = C_{pe,1} + (C_{pe,10} - C_{pe,1}) \cdot \log_{10} A$$

Zona influencia	Parámetro 6,4m ²	Parámetro 10m ²	Parámetro 1m ²
D	0,839	0,8	1,0
A	-1,239	-1,2	-1,4
E	-0,7	-0,7	-0,7

Y así obtenemos dos valores de c_e

$$c_{e,frente} = 0,839$$

$$c_{e,detrás} = -0,7$$

Según el reglamento, el signo menos indica succión. El criterio de signos viene definido como, para los casos de presión las líneas de fuerza del viento son entrantes a la superficie considerada. En cambio en el caso de succión las flechas son salientes.

2. Anejos a la memoria

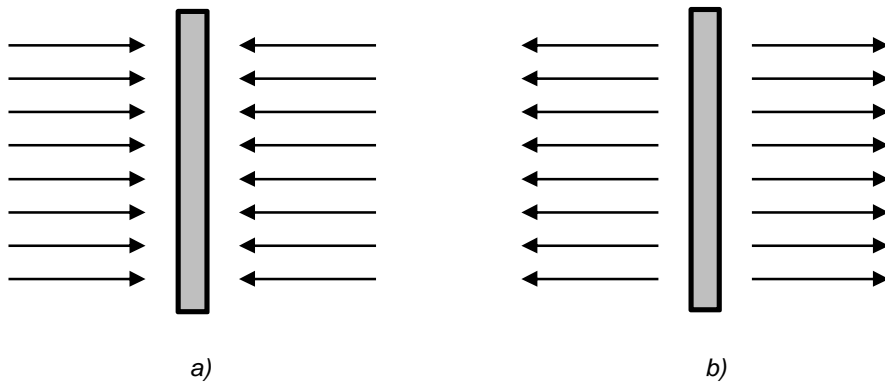


Fig. 49

En nuestro caso tenemos, presión en la cara en al que incide el viento y succión en la cara trasera. Esto último se debe a las turbulencias que se generan en la parte trasera de la placa.

Vamos a obtener la fuerza por unidad de superficie en cada cara:

$$q_e = q_b \cdot c_e \cdot c_p$$
$$q_b = \frac{0,42kN}{m^2}, c_e = 2,0$$
$$q_e = 0,42 \cdot 2 \cdot c_p$$

Calculamos los parámetros para cada tipo de viento:

$$c_{e,frente} = 0,839$$
$$c_{e,detrás} = -0,7$$

Y con ellos obtenemos las acciones finales

$$q_{e,frente} = 0,42 \cdot 2 \cdot 0,839 = 0,7048kN/m^2$$
$$q_{e,detrás} = 0,42 \cdot 2 \cdot (-0,7) = -0,5880kN/m^2$$

2. Anejos a la memoria

De este modo obtenemos los dos casos de viento, que sabemos que no podrán actuar de forma simultánea.

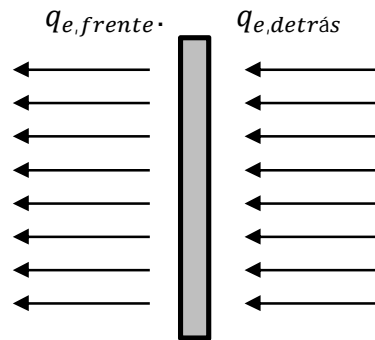


Fig. 50

No obstante, para poder dimensionar la estructura debemos calcular los casos de viento y peso propio para diferentes posiciones para determinar cuál es la más crítica.

Como hacer esto a mano es operativamente inviable, utilizaremos el programa cype que nos permitirá diseñar la estructura teniendo en cuenta las cargas mencionadas. Además del caso de nieve que no hemos tenido en cuenta en el ejemplo realizado a mano debido a que las placas estaban en posición vertical.

2.9.2. Resultados del cálculo

Datos de Obra:

Estructura

La estructura constará de un pilar circular central, sobre este se construirá una barra de perfil tubular cuadrado con aletas, sobre las que se montarán los mecanismos de tres barras para controlar la inclinación de las placas.

Estos mecanismos constarán de una barra de perfil tubular cuadrado, cuyo extremo se deslizará de forma paralela al armazón de sujeción de las placas.

Este armazón constará de dos perfiles IPE arriostrados con correas de perfil en forma de C y además se le añadirán dos montantes de perfil cuadrado para rigidizar el conjunto.

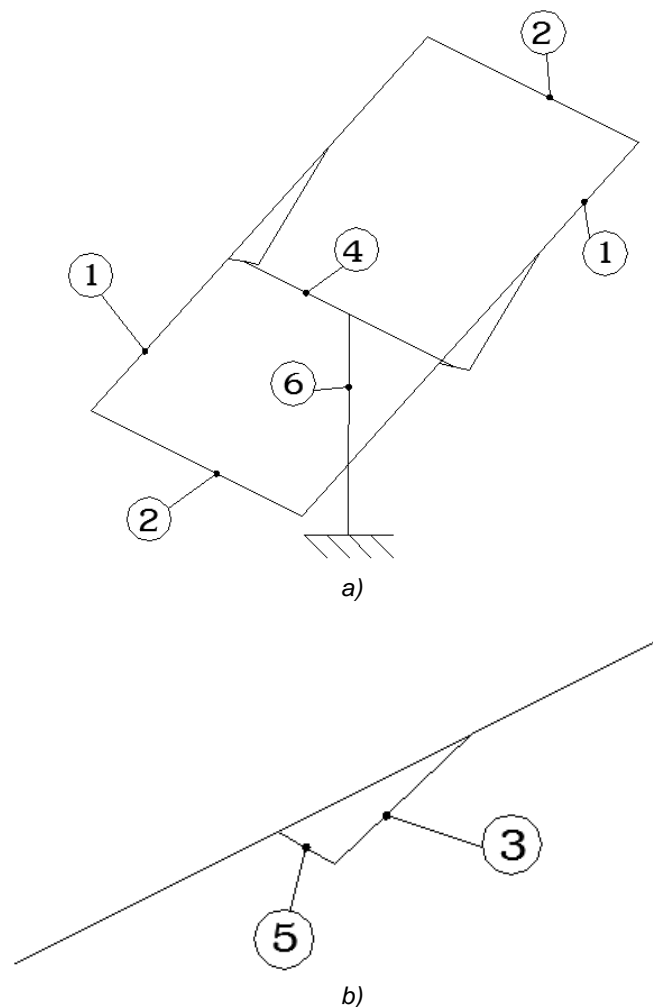


Fig. 51

2. Anejos a la memoria

Marca	Inclinación de las placas					Perfil elegido
	0°	30°	45°	60°	80°	
1	IPE 80	IPE 80	IPE 80	IPE 80	IPE 100	IPE 100
2	TUBO 20 20 2	TUBO 20 20 2	TUBO 20 20 2	TUBO 20 20 2	TUBO 20 20 2	TUBO 20 20 2
3	TUBO 20 20 2	TUBO 30 20 3	TUBO 20 20 2	TUBO 20 20 2	TUBO 20 20 2	TUBO 30 20 3
4	TUBO 90 90 4	TUBO 100 100 5	TUBO 150 50 4	TUBO 120 60 4	TUBO 140 70 5	TUBO 120 120 5
5	PL 60 8	PL 70 8	PL 60 8	PL 60 8	PL 60 8	PL 120 8
6	CHS 70 3	CHS 125 4.5	CHS 150 4.5	CHS 200 4.8	CHS 200 4.8	CHS 200 4.8

El caso de la marca 1 hemos elegido el IPE 100 por ser el que cumple todas las posiciones posibles.

En el caso de la marca 3 y 6, otra vez hemos seguido el criterio de seleccionar el perfil correspondiente a la situación más desfavorable.

El caso de la marca 4 es un poco más complejo, puesto que para cada posición las proporciones de la sección cambian y resulta difícil seleccionar un perfil.

Para llevar a cabo esta decisión, calculamos la estructura en cada una de las 5 posiciones con los perfiles seleccionados para las marcas 1, 2, 3, 5 y 6. En el caso de la marca 5 se seleccionó el perfil PL 70 8 por ser el más desfavorable.

El resultado del cálculo mostró que el TUBO 140 70 5 cumplía en todas las posiciones. No obstante en algunos casos tenía un aprovechamiento cercano al 95%

Se optó por el perfil siguiente para estar del lado de la seguridad.

2. Anejos a la memoria

Debido a que las marcas 4 y 5 se unirán mediante soldadura, se ha considerado apropiado escoger un PL 120 8 para el perfil 5, de este modo el canto de ambas piezas coincidirá y el conjunto será más firme. Como puede verse en la figura: (o ver Plano 5)

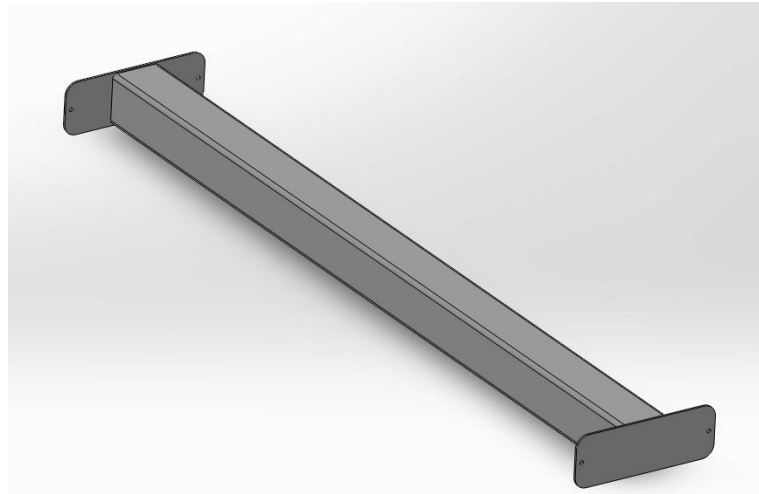


Fig. 52

Elemento	Tipo perfil	Material	Longitud (m)	Volumen (cm ³)	Peso (kg)
N1/N2	IPE 100	Acero S275	3,2	3302,4	25,92
N3/N4	IPE 100		3,2	3302,4	25,92
N2/N4	TUBO 20 20 2		2	266	2,09
N1/N3	TUBO 20 20 2		2	266	2,09
N6/N9	TUBO 30 20 3		0,93	223,2	1,75
N8/N10	TUBO 30 20 3		0,93	223,2	1,75
N11/N12	TUBO 120 120 5		2	4468	35,07
N5/N6	PL 120 8		0,33	316,8	2,49
N7/N8	PL 120 8		0,33	316,8	2,49
N13/N14	CHS 200 4,8		1,6	5161,6	40,52
Correa 1	UL 65 42		2	1640	12,81
Correa 2	UL 65 42		2	1640	12,81
Correa 3	UL 65 42		2	1640	12,81
Correa 4	UL 65 42		2	1640	12,81

Combinaciones de esfuerzos

Para las distintas situaciones de proyecto, las combinaciones de acciones se definirán de acuerdo con los siguientes criterios:

- Con coeficientes de combinación

$$\sum_{j \geq 1} \gamma_{Gj} G_{kj} + \gamma_P P_k + \gamma_{Q1} \Psi_{p1} Q_{k1} + \sum_{i > 1} \gamma_{Qi} \Psi_{ai} Q_{ki}$$

- Sin coeficientes de combinación

$$\sum_{j \geq 1} \gamma_{Gj} G_{kj} + \gamma_P P_k + \sum_{i \geq 1} \gamma_{Qi} Q_{ki}$$

- Donde:

G_k Acción permanente

P_k Acción de pretensado

Q_k Acción variable

γ_G Coeficiente parcial de seguridad de las acciones permanentes

γ_P Coeficiente parcial de seguridad de la acción de pretensado

γ_{Q,1} Coeficiente parcial de seguridad de la acción variable principal

γ_{Q,i} Coeficiente parcial de seguridad de las acciones variables de acompañamiento

γ_{p,1} Coeficiente de combinación de la acción variable principal

γ_{a,i} Coeficiente de combinación de las acciones variables de acompañamiento

Para cada situación de proyecto y estado límite los coeficientes a utilizar serán:
E.L.U. de rotura. Hormigón en cimentaciones: EHE-98-CTE

Persistente o transitoria				
	Coeficientes parciales de seguridad (□)		Coeficientes de combinación (□)	
	Favorable	Desfavorable	Principal (□ _p)	Acompañamiento (□ _a)
Carga permanente (G)	1,000	1,600	-	-
Viento (Q)	0,000	1,600	1,000	0,600
Nieve (Q)	0,000	1,600	1,000	0,500

2. Anejos a la memoria

E.L.U. de rotura. Acero laminado: CTE DB SE-A

Persistente o transitoria				
	Coeficientes parciales de seguridad (γ)		Coeficientes de combinación (ψ)	
	Favorable	Desfavorable	Principal (ψ_p)	Acompañamiento (ψ_a)
Carga permanente (G)	0,800	1,350	-	-
Viento (Q)	0,000	1,500	1,000	0,600
Nieve (Q)	0,000	1,500	1,000	0,500

Tensiones sobre el terreno

Acciones variables sin sismo		
	Coeficientes parciales de seguridad (γ)	
	Favorable	Desfavorable
Carga permanente (G)	1,000	1,000
Viento (Q)	0,000	1,000
Nieve (Q)	0,000	1,000

Desplazamientos

Acciones variables sin sismo		
	Coeficientes parciales de seguridad (γ)	
	Favorable	Desfavorable
Carga permanente (G)	1,000	1,000
Viento (Q)	0,000	1,000
Nieve (Q)	0,000	1,000

Estados de carga

Nombres de las hipótesis

PP Peso propio

V H1 Cubiertas aisladas

V H2 Cubiertas aisladas

N(EI) Nieve (estado inicial)

N(R) Nieve (redistribución)

2. Anejos a la memoria

E.L.U. de rotura. Hormigón en cimentaciones

Comb.	PP	V H1	V H2	N(EI)	N(R)
1	1,000				
2	1,600				
3	1,000	1,600			
4	1,600	1,600			
5	1,000		1,600		
6	1,600		1,600		
7	1,000			1,600	
8	1,600			1,600	
9	1,000	0,960		1,600	
10	1,600	0,960		1,600	
11	1,000		0,960	1,600	
12	1,600		0,960	1,600	
13	1,000	1,600		0,800	
14	1,600	1,600		0,800	
15	1,000		1,600	0,800	
16	1,600		1,600	0,800	
17	1,000				1,600
18	1,600				1,600
19	1,000	0,960			1,600
20	1,600	0,960			1,600
21	1,000		0,960		1,600
22	1,600		0,960		1,600
23	1,000	1,600			0,800
24	1,600	1,600			0,800
25	1,000		1,600		0,800
26	1,600		1,600		0,800

2. Anejos a la memoria

E.L.U. de rotura. Acero laminado

Comb.	PP	V H1	V H2	N(EI)	N(R)
1	0,800				
2	1,350				
3	0,800	1,500			
4	1,350	1,500			
5	0,800		1,500		
6	1,350		1,500		
7	0,800			1,500	
8	1,350			1,500	
9	0,800	0,900		1,500	
10	1,350	0,900		1,500	
11	0,800		0,900	1,500	
12	1,350		0,900	1,500	
13	0,800	1,500		0,750	
14	1,350	1,500		0,750	
15	0,800		1,500	0,750	
16	1,350		1,500	0,750	
17	0,800				1,500
18	1,350				1,500
19	0,800	0,900			1,500
20	1,350	0,900			1,500
21	0,800		0,900		1,500
22	1,350		0,900		1,500
23	0,800	1,500			0,750
24	1,350	1,500			0,750
25	0,800		1,500		0,750
26	1,350		1,500		0,750

Comprobaciones ELU. Flecha

Referencias:

Pos.: Valor de la coordenada sobre el eje 'X' local del grupo de flecha en el punto donde se produce el valor pésimo de la flecha.

L.: Distancia entre dos puntos de corte consecutivos de la deformada con la recta que une los nudos extremos del grupo de flecha.

Flechas									
Grupo	Flecha máxima absoluta xy		Flecha máxima absoluta xz		Flecha activa absoluta xy		Flecha activa absoluta xz		
	Flecha máxima relativa xy		Flecha máxima relativa xz		Flecha activa relativa xy		Flecha activa relativa xz		
	Pos. (m)	Flecha (mm)	Pos. (m)	Flecha (mm)	Pos. (m)	Flecha (mm)	Pos. (m)	Flecha (mm)	
N1/N 2	1,635	0,55	1,635	6,51	1,635	1,08	1,635	12,81	
	1,635	L/(>1000)	1,635	L/550,0	1,635	L/(>1000)	1,635	L/618,0	
N3/N 4	1,635	0,55	1,635	6,51	1,635	1,08	1,635	12,81	
	1,635	L/(>1000)	1,635	L/550,0	1,635	L/(>1000)	1,635	L/618,0	
N8/N 7	0,165	0,01	0,165	0,02	0,165	0,03	0,165	0,04	
	0,165	L/(>1000)	0,165	L/(>1000)	0,165	L/(>1000)	0,165	L/(>1000)	
N6/N 5	0,165	0,01	0,165	0,02	0,165	0,03	0,165	0,04	
	0,165	L/(>1000)	0,165	L/(>1000)	0,165	L/(>1000)	0,165	L/(>1000)	
N8/N 10	0,465	0,00	0,465	0,03	0,465	0,00	0,465	0,00	
	-	L/(>1000)	0,465	L/(>1000)	-	L/(>1000)	-	L/(>1000)	
N6/N 9	0,465	0,00	0,465	0,03	0,465	0,00	0,465	0,00	
	-	L/(>1000)	0,465	L/(>1000)	-	L/(>1000)	-	L/(>1000)	
N11/ N12	1,000	1,86	1,000	2,37	1,000	3,22	1,000	4,69	
	1,000	L/(>1000)	1,000	L/843,0	1,000	L/(>1000)	1,000	L/919,1	
N14/ N13	0,500	0,00	0,500	0,26	0,500	0,00	0,500	0,49	
	-	L/(>1000)	0,500	L/(>1000)	-	L/(>1000)	0,500	L/(>1000)	
N2/N 4	1,000	0,66	1,000	1,36	1,000	0,38	1,000	0,19	
	1,000	L/(>1000)	1,000	L/(>1000)	1,000	L/(>1000)	1,000	L/(>1000)	
N1/N 3	1,000	0,68	1,000	1,37	1,000	0,43	1,000	0,21	
	1,000	L/(>1000)	1,000	L/(>1000)	1,000	L/(>1000)	1,000	L/(>1000)	

Comprobaciones ELU. Ejemplo una barra.

Para mostrar el proceso de cálculo para cada barra, hemos incluido el ejemplo de cálculo de una de las barras móviles (N8/N10) que son las que tienen un mayor aprovechamiento:

Barra N8/N10

Perfil: TUBO 30*20*3 Material: Acero (S275)						
Nudos		Longitud (m)	Características mecánicas			
Inicial	Final		Área (cm ²)	I _y ⁽¹⁾ (cm ⁴)	I _z ⁽¹⁾ (cm ⁴)	I _t ⁽²⁾ (cm ⁴)
N8	N10	0,930	2,40	2,36	1,21	2,95
Notas: (1) Inercia respecto al eje indicado (2) Momento de inercia a torsión uniforme						
		Pandeo		Pandeo lateral		
		Plano XY	Plano XZ	Ala sup.	Ala inf.	
β		1,00	1,00	0,00	0,00	
L _k		0,930	0,930	0,000	0,000	
C _m		1,000	1,000	1,000	1,000	
C ₁		-	-	1,000		
Notación: β: Coeficiente de pandeo L _k : Longitud de pandeo (m) C _m : Coeficiente de momentos C ₁ : Factor de modificación para el momento crítico						

Limitación de esbeltez (CTE DB SE-A, Artículos 6.3.1 y 6.3.2.1 - Tabla 6.3)

La esbeltez reducida $\bar{\lambda}$ de las barras comprimidas debe ser inferior al valor 2.0.

$$\bar{\lambda} = \sqrt{\frac{A \cdot f_y}{N_{cr}}}$$

$$\bar{\lambda} : \underline{1.51} \quad \checkmark$$

Donde:

Clase: Clase de la sección, según la capacidad de deformación y de desarrollo de la resistencia plástica de los elementos planos comprimidos de una sección. Clase : 1

A: Área de la sección bruta para las secciones de clase 1, 2 y 3.

$$A : \underline{2.40} \text{ cm}^2$$

f_y: Límite elástico. (CTE DB SE-A, Tabla 4.1)

$$f_y : \underline{2803.26} \text{ kp/cm}^2$$

N_{cr}: Axil crítico de pandeo elástico.

$$N_{cr} : \underline{2.967} \text{ t}$$

El axil crítico de pandeo elástico N_{cr} es el menor de los valores obtenidos en a), b) y c):

a) Axil crítico elástico de pandeo por flexión respecto al eje Y.

$$N_{cr,y} : \underline{5.768} \text{ t}$$

$$N_{cr,y} = \frac{\pi^2 \cdot E \cdot I_y}{L_{ky}^2}$$

2. Anejos a la memoria

b) Axil crítico elástico de pandeo por flexión respecto al eje Z.

$$N_{cr,z} : \underline{2.967} \text{ t}$$

$$N_{cr,z} = \frac{\pi^2 \cdot E \cdot I_z}{L_{kz}^2}$$

c) Axil crítico elástico de pandeo por torsión.

$$N_{cr,T} : \underline{\infty}$$

$$N_{cr,T} = \frac{1}{i_0^2} \cdot \left[G \cdot I_t + \frac{\pi^2 \cdot E \cdot I_w}{L_{kt}^2} \right]$$

Donde:

I_y : Momento de inercia de la sección bruta, respecto al eje Y.

$$I_y : \underline{2.36} \text{ cm}^4$$

I_z : Momento de inercia de la sección bruta, respecto al eje Z.

$$I_z : \underline{1.21} \text{ cm}^4$$

I_t : Momento de inercia a torsión uniforme.

$$I_t : \underline{2.95} \text{ cm}^4$$

I_w : Constante de alabeo de la sección.

$$I_w : \underline{0.00} \text{ cm}^6$$

E: Módulo de elasticidad.

$$E : \underline{2140673} \text{ kp/cm}^2$$

G: Módulo de elasticidad transversal.

$$G : \underline{825688} \text{ kp/cm}^2$$

L_{ky} : Longitud efectiva de pandeo por flexión, respecto al eje Y.

$$L_{ky} : \underline{0.930} \text{ m}$$

L_{kz} : Longitud efectiva de pandeo por flexión, respecto al eje Z.

$$L_{kz} : \underline{0.930} \text{ m}$$

L_{kt} : Longitud efectiva de pandeo por torsión.

$$L_{kt} : \underline{0.000} \text{ m}$$

i_0 : Radio de giro polar de la sección bruta, respecto al centro de torsión.

$$i_0 : \underline{1.22} \text{ cm}$$

$$i_0 = (i_y^2 + i_z^2 + y_0^2 + z_0^2)^{0.5}$$

Siendo:

i_y , i_z : Radios de giro de la sección bruta, respecto a los ejes principales de inercia Y y Z.

$$i_y : \underline{0.99} \text{ cm}$$

$$i_z : \underline{0.71} \text{ cm}$$

y_0 , z_0 : Coordenadas del centro de torsión en la dirección de los ejes principales Y y Z, respectivamente, relativas al centro de gravedad de la sección.

$$y_0 : \underline{0.00} \text{ mm}$$

$$z_0 : \underline{0.00} \text{ mm}$$

2. Anejos a la memoria

Abolladura del alma inducida por el ala comprimida (Criterio de CYPE Ingenieros, basado en: Eurocódigo 3 EN 1993-1-5: 2006, Artículo 8)

Se debe satisfacer:

$$\frac{h_w}{t_w} \leq k \frac{E}{f_{yf}} \sqrt{\frac{A_w}{A_{fc,ef}}} \quad 8.00 \leq 354.91 \quad \checkmark$$

Donde:

h_w : Altura del alma.

t_w : Espesor del alma.

A_w : Área del alma.

$A_{fc,ef}$: Área reducida del ala comprimida.

k : Coeficiente que depende de la clase de la sección.

E : Módulo de elasticidad.

f_{yf} : Límite elástico del acero del ala comprimida.

Siendo:

$$f_{yf} = f_y$$

$$h_w : \underline{24.00} \text{ mm}$$

$$t_w : \underline{3.00} \text{ mm}$$

$$A_w : \underline{1.44} \text{ cm}^2$$

$$A_{fc,ef} : \underline{0.60} \text{ cm}^2$$

$$k : \underline{0.30}$$

$$E : \underline{2140673} \text{ kp/cm}^2$$

$$f_{yf} : \underline{2803.26} \text{ kp/cm}^2$$

Resistencia a tracción (CTE DB SE-A, Artículo 6.2.3)

Se debe satisfacer:

$$\eta = \frac{N_{t,Ed}}{N_{t,Rd}} \leq 1 \quad \eta : \underline{0.213} \quad \checkmark$$

El esfuerzo solicitante de cálculo pésimo se produce en el nudo N10, para la combinación de acciones 0.8·PP+1.5·VH2.

$N_{t,Ed}$: Axil de tracción solicitante de cálculo pésimo.

$$N_{t,Ed} : \underline{1.366} \text{ t}$$

La resistencia de cálculo a tracción $N_{t,Rd}$ viene dada por:

$$N_{t,Rd} = A \cdot f_{yd}$$

$$N_{t,Rd} : \underline{6.415} \text{ t}$$

Donde:

A : Área bruta de la sección transversal de la barra.

$$A : \underline{2.40} \text{ cm}^2$$

f_{yd} : Resistencia de cálculo del acero.

$$f_{yd} : \underline{2669.77} \text{ kp/cm}^2$$

$$f_{yd} = f_y / \gamma_{M0}$$

Siendo:

f_y : Límite elástico. (CTE DB SE-A, Tabla 4.1)

$$f_y : \underline{2803.26} \text{ kp/cm}^2$$

γ_{M0} : Coeficiente parcial de seguridad del material.

$$\gamma_{M0} : \underline{1.05}$$

2. Anejos a la memoria

Resistencia a compresión (CTE DB SE-A, Artículo 6.2.5)

Se debe satisfacer:

$$\eta = \frac{N_{c,Ed}}{N_{c,Rd}} \leq 1$$

$$\eta : \underline{0.206} \quad \checkmark$$

$$\eta = \frac{N_{c,Ed}}{N_{b,Rd}} \leq 1$$

$$\eta : \underline{0.659} \quad \checkmark$$

El esfuerzo solicitante de cálculo pésimo se produce en el nudo N8, para la combinación de acciones 1.35·PP+1.5·VH1+0.75·N(EI).

$$N_{c,Ed}: \text{Axil de compresión solicitante de cálculo pésimo.} \quad N_{c,Ed} : \underline{1.320} \quad \text{t}$$

La resistencia de cálculo a compresión $N_{c,Rd}$ viene dada por:

$$N_{c,Rd} = A \cdot f_{yd} \quad N_{c,Rd} : \underline{6.415} \quad \text{t}$$

Donde:

Clase: Clase de la sección, según la capacidad de deformación y de desarrollo de la resistencia plástica de los elementos planos comprimidos de una sección. Clase : 1

A: Área de la sección bruta para las secciones de clase 1, 2 y 3. A : 2.40 cm²

f_{yd} : Resistencia de cálculo del acero. $f_{yd} : \underline{2669.77}$ kp/cm²

$$f_{yd} = f_y / \gamma_{M0}$$

Siendo:

f_y : Límite elástico. (CTE DB SE-A, Tabla 4.1) $f_y : \underline{2803.26}$ kp/cm²

γ_{M0} : Coeficiente parcial de seguridad del material. $\gamma_{M0} : \underline{1.05}$

Resistencia a pandeo: (CTE DB SE-A, Artículo 6.3.2)

La resistencia de cálculo a pandeo $N_{b,Rd}$ en una barra comprimida viene dada por:

$$N_{b,Rd} = \chi \cdot A \cdot f_{yd} \quad N_{b,Rd} : \underline{2.004} \quad \text{t}$$

Donde:

A: Área de la sección bruta para las secciones de clase 1, 2 y 3. A : 2.40 cm²

f_{yd} : Resistencia de cálculo del acero. $f_{yd} : \underline{2669.77}$ kp/cm²

$$f_{yd} = f_y / \gamma_{M1}$$

Siendo:

f_y : Límite elástico. (CTE DB SE-A, Tabla 4.1) $f_y : \underline{2803.26}$ kp/cm²

γ_{M1} : Coeficiente parcial de seguridad del material. $\gamma_{M1} : \underline{1.05}$

χ : Coeficiente de reducción por pandeo.

2. Anejos a la memoria

$$\chi = \frac{1}{\Phi + \sqrt{\Phi^2 - (\bar{\lambda})^2}} \leq 1$$

$$\chi_y : 0.49$$

$$\chi_z : 0.31$$

Siendo:

$$\Phi = 0.5 \cdot \left[1 + \alpha \cdot (\bar{\lambda} - 0.2) + (\bar{\lambda})^2 \right]$$

$$\phi_y : 1.30$$

$$\phi_z : 1.96$$

α : Coeficiente de imperfección elástica.

$$\alpha_y : 0.49$$

$$\alpha_z : 0.49$$

$\bar{\lambda}$: Esbeltez reducida.

$$\bar{\lambda} = \sqrt{\frac{A \cdot f_y}{N_{cr}}}$$

$$\bar{\lambda}_y : 1.08$$

$$\bar{\lambda}_z : 1.51$$

N_{cr} : Axil crítico elástico de pandeo, obtenido como el menor de los siguientes valores:

$$N_{cr} : 2.967 \text{ t}$$

$N_{cr,y}$: Axil crítico elástico de pandeo por flexión respecto al eje Y.

$$N_{cr,y} : 5.768 \text{ t}$$

$N_{cr,z}$: Axil crítico elástico de pandeo por flexión respecto al eje Z.

$$N_{cr,z} : 2.967 \text{ t}$$

$N_{cr,T}$: Axil crítico elástico de pandeo por torsión.

$$N_{cr,T} : \infty$$

Resistencia a flexión eje Y (CTE DB SE-A, Artículo 6.2.6)

Se debe satisfacer:

$$\eta = \frac{M_{Ed}}{M_{c,Rd}} \leq 1$$

$$\eta : 0.003 \quad \checkmark$$

Para flexión positiva:

El esfuerzo solicitante de cálculo pésimo se produce en un punto situado a una distancia de 0.465 m del nudo N8, para la combinación de acciones 1.35-PP.

M_{Ed}^+ : Momento flector solicitante de cálculo pésimo.

$$M_{Ed}^+ : 0.000 \text{ t}\cdot\text{m}$$

Para flexión negativa:

M_{Ed}^- : Momento flector solicitante de cálculo pésimo.

$$M_{Ed}^- : 0.000 \text{ t}\cdot\text{m}$$

El momento flector resistente de cálculo $M_{c,Rd}$ viene dado por:

$$M_{c,Rd} = W_{pl,y} \cdot f_{yd}$$

$$M_{c,Rd} : 0.066 \text{ t}\cdot\text{m}$$

2. Anejos a la memoria

Donde:

Clase: Clase de la sección, según la capacidad de Clase : 1
deformación y de desarrollo de la resistencia plástica de los elementos planos de una sección a flexión simple.

$W_{pl,y}$: Módulo resistente plástico correspondiente a la fibra $W_{pl,y}$: 2.48 cm³
con mayor tensión, para las secciones de clase 1 y 2.

f_{yd} : Resistencia de cálculo del acero. f_{yd} : 2669.77 kp/cm²

$$f_{yd} = f_y / \gamma_{M0}$$

Siendo:

f_y : Límite elástico. (CTE DB SE-A, Tabla 4.1) f_y : 2803.26 kp/cm²

γ_{M0} : Coeficiente parcial de seguridad del material. γ_{M0} : 1.05

Resistencia a pandeo lateral: (CTE DB SE-A, Artículo 6.3.3.2)
No procede, dado que las longitudes de pandeo lateral son nulas.

Resistencia a flexión eje Z (CTE DB SE-A, Artículo 6.2.6)

La comprobación no procede, ya que no hay momento flector.

Resistencia a corte Z (CTE DB SE-A, Artículo 6.2.4)

Se debe satisfacer:

$$\eta = \frac{V_{Ed}}{V_{c,Rd}} \leq 1$$

$$\eta < \underline{0.001} \quad \checkmark$$

El esfuerzo solicitante de cálculo pésimo se produce en el nudo N8, para la combinación de acciones 1.35·PP.

V_{Ed} : Esfuerzo cortante solicitante de cálculo pésimo. V_{Ed} : 0.001 t

El esfuerzo cortante resistente de cálculo $V_{c,Rd}$ viene dado por:

$$V_{c,Rd} = A_v \cdot \frac{f_{yd}}{\sqrt{3}}$$

$$V_{c,Rd} : \underline{2.220} \text{ t}$$

Donde:

A_v : Área transversal a cortante. A_v : 1.44 cm²

$$A_v = 2 \cdot d \cdot t_w$$

Siendo:

d : Altura del alma. d : 24.00 mm

t_w : Espesor del alma. t_w : 3.00 mm

f_{yd} : Resistencia de cálculo del acero. f_{yd} : 2669.77 kp/cm²

$$f_{yd} = f_y / \gamma_{M0}$$

Siendo:

f_y : Límite elástico. (CTE DB SE-A, Tabla 4.1) f_y : 2803.26 kp/cm²

γ_{M0} : Coeficiente parcial de seguridad del material. γ_{M0} : 1.05

2. Anejos a la memoria

Abolladura por cortante del alma: (CTE DB SE-A, Artículo 6.3.3.4)

Aunque no se han dispuesto rigidizadores transversales, no es necesario comprobar la resistencia a la abolladura del alma, puesto que se cumple:

$$\frac{d}{t_w} < 70 \cdot \varepsilon \quad 8.00 < 64.71 \quad \checkmark$$

Donde:

λ_w : Esbeltez del alma.

$$\lambda_w : \underline{8.00}$$

$$\lambda_w = \frac{d}{t_w}$$

λ_{\max} : Esbeltez máxima.

$$\lambda_{\max} : \underline{64.71}$$

$$\lambda_{\max} = 70 \cdot \varepsilon$$

ε : Factor de reducción.

$$\varepsilon : \underline{0.92}$$

$$\varepsilon = \sqrt{\frac{f_{\text{ref}}}{f_y}}$$

Siendo:

f_{ref} : Límite elástico de referencia.

$$f_{\text{ref}} : \underline{2395.51 \text{ kp/cm}^2}$$

f_y : Límite elástico. (CTE DB SE-A, Tabla 4.1)

$$f_y : \underline{2803.26 \text{ kp/cm}^2}$$

Resistencia a corte Y (CTE DB SE-A, Artículo 6.2.4)

La comprobación no procede, ya que no hay esfuerzo cortante.

Resistencia a momento flector Y y fuerza cortante Z combinados (CTE DB SE-A, Artículo 6.2.8)

No es necesario reducir la resistencia de cálculo a flexión, ya que el esfuerzo cortante solicitante de cálculo pésimo V_{Ed} no es superior al 50% de la resistencia de cálculo a cortante $V_{c,Rd}$.

$$V_{Ed} \leq \frac{V_{c,Rd}}{2} \quad 0.000 \text{ t} \leq 1.110 \text{ t} \quad \checkmark$$

Los esfuerzos solicitantes de cálculo pésimos se producen en un punto situado a una distancia de 0.233 m del nudo N8, para la combinación de acciones 1.35·PP.

V_{Ed} : Esfuerzo cortante solicitante de cálculo pésimo.

$$V_{Ed} : \underline{0.000} \text{ t}$$

$V_{c,Rd}$: Esfuerzo cortante resistente de cálculo.

$$V_{c,Rd} : \underline{2.220} \text{ t}$$

2. Anejos a la memoria

Resistencia a momento flector Z y fuerza cortante Y combinados (CTE DB SE-A, Artículo 6.2.8)

No hay interacción entre momento flector y esfuerzo cortante para ninguna combinación. Por lo tanto, la comprobación no procede.

Resistencia a flexión y axil combinados (CTE DB SE-A, Artículo 6.2.8)

Se debe satisfacer:

$$\eta = \frac{N_{c,Ed}}{N_{pl,Rd}} + \frac{M_{y,Ed}}{M_{pl,Rd,y}} + \frac{M_{z,Ed}}{M_{pl,Rd,z}} \leq 1 \quad \eta : \underline{0.209} \quad \checkmark$$

$$\eta = \frac{N_{c,Ed}}{\chi_y \cdot A \cdot f_{yd}} + k_y \cdot \frac{C_{m,y} \cdot M_{y,Ed}}{\chi_{LT} \cdot W_{pl,y} \cdot f_{yd}} + \alpha_z \cdot k_z \cdot \frac{C_{m,z} \cdot M_{z,Ed}}{W_{pl,z} \cdot f_{yd}} \leq 1 \quad \eta : \underline{0.420} \quad \checkmark$$

$$\eta = \frac{N_{c,Ed}}{\chi_z \cdot A \cdot f_{yd}} + \alpha_y \cdot k_y \cdot \frac{C_{m,y} \cdot M_{y,Ed}}{W_{pl,y} \cdot f_{yd}} + k_z \cdot \frac{C_{m,z} \cdot M_{z,Ed}}{W_{pl,z} \cdot f_{yd}} \leq 1 \quad \eta : \underline{0.661} \quad \checkmark$$

Los esfuerzos solicitantes de cálculo pésimos se producen en un punto situado a una distancia de 0.465 m del nudo N8, para la combinación de acciones 1.35·PP+1.5·VH1+0.75·N(EI).

Donde:

$N_{c,Ed}$: Axil de compresión solicitante de cálculo pésimo. $N_{c,Ed} : \underline{1.319} \text{ t}$

$M_{y,Ed}$, $M_{z,Ed}$: Momentos flectores solicitantes de cálculo pésimos, según los ejes Y y Z, respectivamente. $M_{y,Ed}^+ : \underline{0.000} \text{ t}\cdot\text{m}$

$M_{z,Ed}^+ : \underline{0.000} \text{ t}\cdot\text{m}$

Clase: Clase de la sección, según la capacidad de deformación y de desarrollo de la resistencia plástica de sus elementos planos, para axil y flexión simple. Clase : 1

$N_{pl,Rd}$: Resistencia a compresión de la sección bruta. $N_{pl,Rd} : \underline{6.415} \text{ t}$

$M_{pl,Rd,y}$, $M_{pl,Rd,z}$: Resistencia a flexión de la sección bruta en condiciones plásticas, respecto a los ejes Y y Z, respectivamente. $M_{pl,Rd,y} : \underline{0.066} \text{ t}\cdot\text{m}$

$M_{pl,Rd,z} : \underline{0.049} \text{ t}\cdot\text{m}$

Resistencia a pandeo: (CTE DB SE-A, Artículo 6.3.4.2)

A: Área de la sección bruta. A : 2.40 cm²

$W_{pl,y}$, $W_{pl,z}$: Módulos resistentes plásticos correspondientes a la fibra comprimida, alrededor de los ejes Y y Z, respectivamente. $W_{pl,y} : \underline{2.48} \text{ cm}^3$

$W_{pl,z} : \underline{1.82} \text{ cm}^3$

f_{yd} : Resistencia de cálculo del acero. $f_{yd} : \underline{2669.77} \text{ kp/cm}^2$

$$f_{yd} = f_y / \gamma_{M1}$$

Siendo:

f_y : Límite elástico. (CTE DB SE-A, Tabla 4.1) $f_y : \underline{2803.26} \text{ kp/cm}^2$

γ_{M1} : Coeficiente parcial de seguridad del material. $\gamma_{M1} : \underline{1.05}$

2. Anejos a la memoria

k_y, k_z : Coeficientes de interacción.

$$k_y = 1 + (\bar{\lambda}_y - 0.2) \cdot \frac{N_{c,Ed}}{\chi_y \cdot N_{c,Rd}} \quad k_y : \underline{1.33}$$

$$k_z = 1 + (\bar{\lambda}_z - 0.2) \cdot \frac{N_{c,Ed}}{\chi_z \cdot N_{c,Rd}} \quad k_z : \underline{1.53}$$

$C_{m,y}, C_{m,z}$: Factores de momento flector uniforme equivalente. $C_{m,y} : \underline{1.00}$
 $C_{m,z} : \underline{1.00}$

χ_y, χ_z : Coeficientes de reducción por pandeo, alrededor de los ejes Y y Z, respectivamente. $\chi_y : \underline{0.49}$
 $\chi_z : \underline{0.31}$

$\bar{\lambda}_y, \bar{\lambda}_z$: Esbelteces reducidas con valores no mayores que 1.00, en relación a los ejes Y y Z, respectivamente. $\bar{\lambda}_y : \underline{1.08}$
 $\bar{\lambda}_z : \underline{1.51}$

α_y, α_z : Factores dependientes de la clase de la sección. $\alpha_y : \underline{0.60}$
 $\alpha_z : \underline{0.60}$

Resistencia a flexión, axil y cortante combinados (CTE DB SE-A, Artículo 6.2.8)

No es necesario reducir las resistencias de cálculo a flexión y a axil, ya que se puede ignorar el efecto de abolladura por esfuerzo cortante y, además, el esfuerzo cortante solicitante de cálculo pésimo V_{Ed} es menor o igual que el 50% del esfuerzo cortante resistente de cálculo $V_{c,Rd}$.

Los esfuerzos solicitantes de cálculo pésimos se producen en un punto situado a una distancia de 0.233 m del nudo N8, para la combinación de acciones 1.35·PP.

$$V_{Ed,z} \leq \frac{V_{c,Rd,z}}{2} \quad 0.000 \text{ t} \leq 1.110 \text{ t} \quad \checkmark$$

Donde:

$V_{Ed,z}$: Esfuerzo cortante solicitante de cálculo pésimo. $V_{Ed,z} : \underline{0.000} \text{ t}$
 $V_{c,Rd,z}$: Esfuerzo cortante resistente de cálculo. $V_{c,Rd,z} : \underline{2.220} \text{ t}$

Resistencia a torsión (CTE DB SE-A, Artículo 6.2.7)

La comprobación no procede, ya que no hay momento torsor.

Resistencia a cortante Z y momento torsor combinados (CTE DB SE-A, Artículo 6.2.8)

No hay interacción entre momento torsor y esfuerzo cortante para ninguna combinación. Por lo tanto, la comprobación no procede.

Resistencia a cortante Y y momento torsor combinados (CTE DB SE-A, Artículo 6.2.8)

No hay interacción entre momento torsor y esfuerzo cortante para ninguna combinación. Por lo tanto, la comprobación no procede.

2.9.3.Cimentación

Descripción

Referencias	Geometría	Armado
N14	Zapata rectangular excéntrica Ancho inicial X: 112,5 cm Ancho inicial Y: 112,5 cm Ancho final X: 112,5 cm Ancho final Y: 112,5 cm Ancho zapata X: 225,0 cm Ancho zapata Y: 225,0 cm Canto: 50,0 cm	Sup X: 9Ø12c/25 Sup Y: 9Ø12c/25 Inf X: 9Ø12c/25 Inf Y: 9Ø12c/25

Medición

Referencia: N14		B 500 S, CN	Total
Nombre de armado		Ø12	
Parrilla inferior - Armado X	Longitud (m)	9x2,15	19,35
	Peso (kg)	9x1,91	17,18
Parrilla inferior - Armado Y	Longitud (m)	9x2,15	19,35
	Peso (kg)	9x1,91	17,18
Parrilla superior - Armado X	Longitud (m)	9x2,15	19,35
	Peso (kg)	9x1,91	17,18
Parrilla superior - Armado Y	Longitud (m)	9x2,15	19,35
	Peso (kg)	9x1,91	17,18
Totales	Longitud (m)	77,40	
	Peso (kg)	68,72	68,72
Total con mermas (10.00%)	Longitud (m)	85,14	
	Peso (kg)	75,59	75,59

Resumen de medición (se incluyen mermas de acero)

Elemento	B 500 S, CN (kg)	Hormigón (m³)	
	Ø12	HA-30, Control Estadístico	Limpieza
Referencia: N14	75,59	2,53	0,51
Totales	75,59	2,53	0,51

2. Anejos a la memoria

Comprobación

Referencia: N14		
Dimensiones: 225 x 225 x 50		
Armados: Xi:Ø12c/25 Yi:Ø12c/25 Xs:Ø12c/25 Ys:Ø12c/25		
Comprobación	Valores	Estado
Tensiones sobre el terreno: Criterio de CYPE Ingenieros		
- Tensión media en situaciones persistentes:	Máximo: 2 kp/cm ² Calculado: 0,158 kp/cm ²	Cumple
- Tensión máxima en situaciones persistentes sin viento:	Máximo: 2,5 kp/cm ² Calculado: 0,136 kp/cm ²	Cumple
- Tensión máxima en situaciones persistentes con viento:	Máximo: 2,5 kp/cm ² Calculado: 0,23 kp/cm ²	Cumple
Vuelco de la zapata:		
- En dirección X (1)		No procede
- En dirección Y: Si el % de reserva de seguridad es mayor que cero, quiere decir que los coeficientes de seguridad al vuelco son mayores que los valores estrictos exigidos para todas las combinaciones de equilibrio.	Reserva seguridad: 68,4 %	Cumple
(1) Sin momento de vuelco		
Flexión en la zapata:		
- En dirección X:	Momento: 0,56 t·m	Cumple
- En dirección Y:	Momento: -1,54 t·m	Cumple
Cortante en la zapata:		
- En dirección X:	Cortante: 0,64 t	Cumple
- En dirección Y:	Cortante: 1,82 t	Cumple
Compresión oblicua en la zapata:		
- Situaciones persistentes: Criterio de CYPE Ingenieros	Máximo: 611,62 t/m ² Calculado: 4,92 t/m ²	Cumple
Canto mínimo: Artículo 59.8.1 de la norma EHE-98	Mínimo: 25 cm Calculado: 50 cm	Cumple
Espacio para anclar arranques en cimentación:		
- N14:	Mínimo: 35 cm Calculado: 43 cm	Cumple

2. Anejos a la memoria

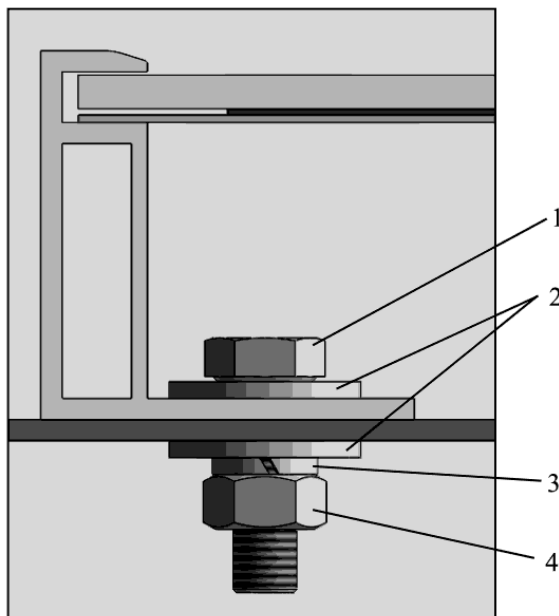
Referencia: N14		
Dimensiones: 225 x 225 x 50		
Armados: Xi:Ø12c/25 Yi:Ø12c/25 Xs:Ø12c/25 Ys:Ø12c/25		
Comprobación	Valores	Estado
Cuantía geométrica mínima: Criterio de CYPE Ingenieros	Mínimo: 0,0018	
- En dirección X:	Calculado: 0,0018	Cumple
- En dirección Y:	Calculado: 0,0018	Cumple
Cuantía mínima necesaria por flexión: Artículo 42.3.2 de la norma EHE-98	Calculado: 0,001	
- Armado inferior dirección X:	Mínimo: 0,0001	Cumple
- Armado inferior dirección Y:	Mínimo: 0,0002	Cumple
- Armado superior dirección X:	Mínimo: 0,0001	Cumple
- Armado superior dirección Y:	Mínimo: 0,0002	Cumple
Diámetro mínimo de las barras: Recomendación del Artículo 59.8.2 (norma EHE-98)	Mínimo: 12 mm	
- Parrilla inferior:	Calculado: 12 mm	Cumple
- Parrilla superior:	Calculado: 12 mm	Cumple
Separación máxima entre barras: Artículo 59.8.2 de la norma EHE-98	Máximo: 30 cm	
- Armado inferior dirección X:	Calculado: 25 cm	Cumple
- Armado inferior dirección Y:	Calculado: 25 cm	Cumple
- Armado superior dirección X:	Calculado: 25 cm	Cumple
- Armado superior dirección Y:	Calculado: 25 cm	Cumple
Separación mínima entre barras: Criterio de CYPE Ingenieros, basado en: J. Calavera. "Cálculo de Estructuras de Cimentación". Capítulo 3.16	Mínimo: 10 cm	
- Armado inferior dirección X:	Calculado: 25 cm	Cumple
- Armado inferior dirección Y:	Calculado: 25 cm	Cumple
- Armado superior dirección X:	Calculado: 25 cm	Cumple
- Armado superior dirección Y:	Calculado: 25 cm	Cumple

2. Anejos a la memoria

Referencia: N14		
Dimensiones: 225 x 225 x 50		
Armados: Xi:Ø12c/25 Yi:Ø12c/25 Xs:Ø12c/25 Ys:Ø12c/25		
Comprobación	Valores	Estado
Longitud de anclaje: Criterio del libro "Cálculo de estructuras de cimentación", J. Calavera. Ed. INTEMAC, 1991	Mínimo: 15 cm	
- Armado inf. dirección X hacia der:	Calculado: 52 cm	Cumple
- Armado inf. dirección X hacia izq:	Calculado: 52 cm	Cumple
- Armado inf. dirección Y hacia arriba:	Calculado: 52 cm	Cumple
- Armado inf. dirección Y hacia abajo:	Calculado: 52 cm	Cumple
- Armado sup. dirección X hacia der:	Calculado: 52 cm	Cumple
- Armado sup. dirección X hacia izq:	Calculado: 52 cm	Cumple
- Armado sup. dirección Y hacia arriba:	Calculado: 52 cm	Cumple
- Armado sup. dirección Y hacia abajo:	Calculado: 52 cm	Cumple
Se cumplen todas las comprobaciones		

2.9.4. Fijación de las placas

El fabricante recomienda su instalación mediante unión roscada:



Siendo:

- 1) Tornillo de métrica 6 de acero inoxidable
- 2) Arandela plana de acero inoxidable
- 3) Arandela de compresión de acero inoxidable
- 4) Tuerca hexagonal de acero inoxidable

Fig. 53

2.10. Cálculo de los actuadores

2.10.1. Motor de elevación

Magnitudes necesarias:

d_m : diámetro medio del tornillo

p : Distancia entre filetes, paso. Distancia avanzada por el tornillo por vuelta dada.

φ : ángulo de rozamiento entre el tornillo y la tuerca

Definición ángulo rozamiento: $\varphi = \text{atan}(\mu)$

λ : ángulo de hélice

d_c : diámetro de la cabeza de la tuerca

μ_c : coeficiente rozamiento entre la cabeza de la tuerca y el material al que este enganchada

Los parámetros básicos del tornillo son:

Relación de transmisión (i)

$$i = \frac{\text{Vel. Salida}}{\text{Vel. Entrada}} = \frac{P}{2\pi} = \frac{d_m \text{tg}(\lambda)}{2}$$

Nota: Podemos definir el paso en función del ángulo de hélice como:

$$P = \pi d_m \text{tg}(\lambda)$$

Ventaja mecánica (VM)

$$VM = \frac{\text{Fuerza obtenida}}{\text{Fuerza aportada}} = \frac{1}{\text{tg}(\varphi + \lambda) \frac{d_m}{2} + \mu_c \frac{d_c}{2}}$$

Rendimiento (η)

$$\eta = i \cdot VM = \frac{\text{tg}(\lambda)}{\text{tg}(\varphi + \lambda) + \mu_c \frac{d_c}{d_m}}$$

Cálculo par motor elevación

Con todos estos parámetros podemos calcular el par de arranque del motor.

A partir de la ventaja mecánica, si obtenemos la fuerza necesaria para elevar las placas podemos obtener el par que deberá aportar el motor.

2. Anejos a la memoria

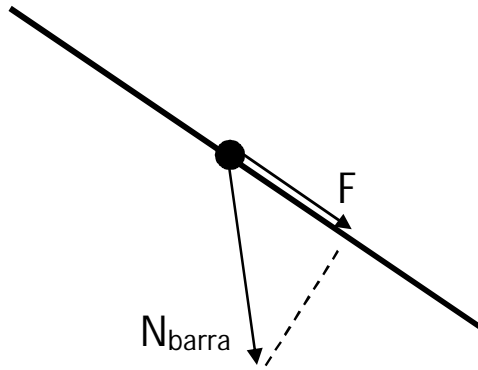


Fig. 54

Y la fuerza necesaria en el husillo para levantar las placas la podemos obtener a partir del axil de la barra móvil, coincidiendo con su componente en la dirección de las placas.

Lo calculamos para varias inclinaciones:

Inclinación de las placas (°)	Ángulo entre la barra y las placas (°)	Carga axial N_{barra} (T)	F (kN)
0	10	0,653	6,30217873
30	18	1,11	10,3455928
45	20	0,665	6,12397681
60	21	0,734	6,71543072
80	19	1,205	11,1656289

Podemos comprobar cómo las posiciones más críticas par el motor será cuando se estén empezando a levantar las placas y cuando estén ya casi verticales.

Obtenemos la ventaja mecánica para nuestro tornillo sinfín. En nuestro caso la fuerza que debemos obtener el F y la fuerza que aportamos es el momento, M:

$$VM = \frac{F}{M} = \frac{1}{\operatorname{tg}(\varphi + \lambda) \frac{d_m}{2} + \mu_c \frac{d_c}{2}}$$

Los datos que necesitamos son:

d_m : diámetro medio del tornillo

p : Distancia entre filetes, paso. Distancia avanzada por el tornillo por vuelta dada.

φ : ángulo de rozamiento entre el tornillo y la tuerca

Definición ángulo rozamiento: $\varphi = \operatorname{atan}(\mu)$

L : avance, distancia que avanza el tornillo en una vuelta

$$L = N p$$

Donde N es el número de filetes de la rosca

2. Anejos a la memoria

λ : ángulo de hélice, que podemos obtener como:

$$\lambda = \text{atan}\left(\frac{L}{\pi d_m}\right)$$

No obstante para obtener estos parámetros primero debemos seleccionar el tornillo.

Vamos a realizar la selección basándonos únicamente en el criterio de resistencia, debido a que gracias al sistema de guiado únicamente soportará tensión axial.

Además como su movimiento será lento, no habrá riesgo de resonancia. Y despreciaremos el pandeo debido a que el tornillo no será demasiado esbelto.

Para calcular la resistencia del husillo debemos obtener el área que este debe tener.

$$A = \frac{F}{P}$$

Siendo F la fuerza que debe resistir y P la presión de contacto, que el fabricante recomienda no sea superior a 5MPa (5N/mm²)

Si aproximamos la fuerza que tenemos a 11,5kN, podemos obtener el área:

$$A_{\text{contacto}} = \frac{F}{P} = \frac{11500N}{5N/mm^2} = 2300 \text{ mm}^2$$

Con ese valor vamos al catálogo y seleccionamos una tuerca que sea capaz de aguantarlo.

En este caso hemos seleccionado una tuerca de bronce de tipo EFM en el catálogo de TECNOPOWER, concretamente el modelo EFM Tr 40x7 que tiene 2930mm² de área de contacto

Con este valor vamos al apartado de husillos y seleccionamos el RPTS Tr 40x7 Cuyos parámetros son:

$$\lambda = 3,483^\circ$$
$$d_m = 36,2\text{mm}$$

Sabiendo esto, podemos obtener el paso:

$$\lambda = \text{atan}\left(\frac{L}{\pi d_m}\right) \rightarrow L = \tan(\lambda) \pi d_m$$

2. Anejos a la memoria

Sustituyendo obtenemos:

$$L = 6.92 \approx 7\text{mm}$$

Puesto que estos modelos solo tienen un filete, el paso es igual al avance.

Y el ángulo de rozamiento, lo podemos calcular sabiendo que el tornillo es de acero y la tuerca de bronce, con el coeficiente de rozamiento mutuo entre acero y bronce.

$$\varphi = \text{atan}(\mu) = \text{atan}(0.15) = 8,5^\circ$$

Finalmente, calculamos la ventaja mecánica, despreciando el rozamiento entre la cabeza de la tuerca y el soporte:

$$VM = \frac{F}{M} = \frac{1}{\text{tg}(\varphi + \lambda) \frac{d_m}{2}} = \frac{1}{\text{tg}(21.8 + 3.483) \frac{36 \cdot 10^{-3}}{2}} = 261,75m$$

Por tanto el par que deberá proporcionar el motor al husillo es:

$$M = \frac{F}{VM} = \frac{11500N}{261,75m} = 43Nm$$

No obstante los motores paso a paso comerciales no suelen ir más allá de los 12,5Nm

Resultado

Si aplicamos una etapa de reducción de 1/20 y le suponemos un rendimiento del 75% con 2,87Nm bastaría.

Hay que tener en cuenta que la aproximación del rendimiento bastante restrictiva, ya que la mayoría de reductoras tienen rendimientos entorno el 80%, por tanto escogeremos un motor de 3,1Nm

Concretamente un Nema23 M23-3.1-1D10-D2-A

2.10.2. Motor de rotación.

Para calcularlo necesitaremos obtener el momento de inercia del seguidor. El proceso de cálculo no es muy complejo, pero sí largo y tedioso ya que hay que calcular el momento de inercia de cada barra y su distancia al eje de giro del seguidor.

2. Anejos a la memoria

No obstante, y puesto que hemos modelado el seguidor completo en SolidWorks, aprovecharemos el trabajo hecho para obtener el momento de inercia. SolidWorks nos proporciona el tensor de inercia respecto el centro de gravedad de la pieza y respecto los ejes globales del sistema. En este caso tomaremos el tursor respecto los ejes globales cuando las placas están horizontales, por ser el caso con mayor inercia:

$$J_{total} = \begin{bmatrix} 464,72 & -105,11 & 0 \\ -105,11 & 384,34 & 0,236 \\ 0 & 0,236 & 486,39 \end{bmatrix} Kg m^2$$

En nuestro caso ensamblamos el modelo colocando el centro de la base del mástil en el origen de coordenadas, de este modo el eje de giro del seguidor coincide con el eje y de las coordenadas globales.

$$I_v = v \cdot J_{total}$$

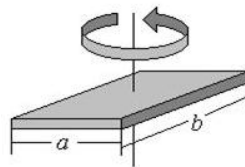
Dónde v es el vector unitario de la dirección respecto de la cual se quiere calcular el momento de inercia. En nuestro caso:

$$I_y = (0, 1, 0) \begin{bmatrix} 464,72 & -105,11 & 0 \\ -105,11 & 384,34 & 0,236 \\ 0 & 0,236 & 486,39 \end{bmatrix} (0, 1, 0)^T = 384.34 Kg m^2$$

En nuestro caso esta operación no tiene demasiada relevancia, ya que los elementos de la diagonal principal coinciden con los momentos de inercia del conjunto respecto cada uno de los ejes principales, y en nuestro caso el eje y se corresponde con una de las direcciones principales.

No obstante, el modelo no incluye las placas. Vamos a calcular su momento de inercia a partir de la fórmula para una placa cuadrada de dimensiones a x b:

$$I_{CM} = \frac{1}{12} M (a^2 + b^2)$$



Placa rectangular

Fig. 55

2. Anejos a la memoria

En nuestro caso consideraremos el conjunto de las 4 placas (990 x 1650mm) como una única plancha con dimensiones:

$$\begin{aligned}a &= 2 \times 0,99 = 1,98m \\ b &= 2 \times 1,65 = 3,3m\end{aligned}$$

Y M será la masa total de las cuatro placas:

$$M = 4 \times 20 = 80Kg$$

Ya podemos calcular el momento de inercia del conjunto de las placas, como:

$$I_{placas} = \frac{1}{12} M (a^2 + b^2) = \frac{1}{12} 80 (1,98^2 + 3,3^2) = 98,74 Kg m^2$$

Finalmente, el momento de inercia total será la suma de ambos, puesto que están calculados respecto el mismo eje:

$$I_{total} = I_y + I_{placas} = 384,34 + 98,74 = 483,1 Kg m^2$$

Podemos relacionar la potencia de salida del motor con la potencia aportada al heliostato como:

$$H = M_{motor} \omega_{motor} = M_{helio} \omega_{helio}$$

Por exigencias de precisión del software de control, la relación de transmisión debe ser del orden de 1/200 debido a que el paso angular del motor es de solo 1,8° y el heliostato deberá girar unas décimas de grado por minuto.

Con lo cual obtenemos la relación:

$$\omega_{motor} = 200 \omega_{helio} \rightarrow 200 M_{motor} = M_{helio}$$

Una reducción así, la forma más compacta de conseguirla es mediante un engranaje de tornillo sinfín. Que además suele ser una transmisión con autorretención. Aunque suelen presentar un rendimiento bajo, por ello tendremos que tener en cuenta el rendimiento.

Si suponemos un rendimiento del 50%

$$M_{motor} = \frac{M_{helio}}{200 \cdot 0,5}$$

Finalmente, podemos calcular el par necesario en el giro del heliostato a partir del momento de inercia de giro como:

2. Anejos a la memoria

$$M_{helio} = I_{total} \alpha$$

Dónde α es la aceleración angular (rad/s^2)

Si tomamos una aceleración de 1rad/s^2 obtenemos:

$$M_{helio} = I_{total} \alpha = 483,1 \cdot 1 = 483,1 \text{ Nm}$$

Para calcular el par del motor necesitamos definir la transmisión. Para ello vamos al catalogo de tornillos sinfín y engranajes.

Hemos elegido uno de diámetro interior 250mm, lo cual permitirá su montaje en el propio tubo. Concretamente el modelo AGF6-60R1 de la marca KHK y su piñón correspondiente, el KWG6R1

Su relación de transmisión es de 1/60 lo cual no es suficiente, así que le acoplaremos una reductora de 1/4 consiguiendo una reducción total de 1/240

Resultado

Con el par necesario y la reducción ya podemos obtener el par motor:

$$M_{motor} = \frac{M_{helio}}{240 \cdot 0,5} = \frac{483,1}{120} = 4,025 \text{ Nm}$$

Para poder aportar ese par, hemos elegido un motor Nema34 de 4.2 Nm de par motor, concretamente el modelo M34-4.2-1SB12-E2-A

3. Planos

Listado de planos:

- Plano 1. Plano de conjunto
- Plano 2. Dimensiones principales del seguidor
- Plano 3. Vista normal del bastidor
- Plano 4. Pieza para el mástil CHS 200
- Plano 5. Tubo transversal. Tubo 120x120 + 2xPL 120x8
- Plano 6. Detalle del sistema de elevación
- Plano 7. Deslizadera (sistema de guiado)
- Plano 8. Guía 600mm
- Plano 9. Placa uniones trasera
- Plano 10. Volumen de trabajo
- Plano 11. Placa de anclaje
- Plano 12. Zapata y mallazo

Listado de datasheets:

- Datasheet 1. Tuerca EFM del husillo (mecanismo de elevación)
- Datasheet 2. Husillo (mecanismo de elevación)
- Datasheet 3. Engranaje con sinfín (mecanismo de rotación)
- Datasheet 4. Motor de elevación Nema 23
- Datasheet 5. Motor de rotación Nema 34
- Datasheet 6. Manual circuito integrado L293/L293D

Todos los documentos de encuentran al final del proyecto en el orden establecido en este apartado.

3. Planos

4. Pliego de condiciones

4.1. Paneles Solares.

Descripción.

El número de paneles situados sobre el seguidor no podrá ser superior a cuatro. Los paneles deben ser colocados de forma tal que no sobresalgan o sobresalgan lo mínimo posible del contorno del soporte de elevación.

Para el diseño del presente proyecto se han tomado como referencia paneles de 1x1,6m. En caso de colocar un tamaño superior al mencionado, es aceptable que su tamaño exceda hasta en un 10% como máximo las magnitudes mencionadas.

Especificaciones.

Los módulos deberán satisfacer las especificaciones UNE-EN 61215, para módulos de silicio cristalino o UNE-EN 61646, para módulos fotovoltaicos de capa delgada, así como estar cualificados por algún laboratorio reconocido y deberá ser acreditado mediante la presentación del certificado correspondiente.

El módulo llevará de forma visible el modelo, nombre, o logotipo del fabricante, así como el número de serie que permita conocer la fecha de fabricación para permitir su identificación individual.

El fabricante deberá proporcionar la información necesaria a la curva de degradación de los paneles, por la cual van perdiendo capacidad de generación de energía con el paso de los años.

Los módulos deben satisfacer las características mecánicas proporcionadas por el fabricante. Prestando especial atención a sus valores de:

Flecha máxima soportada por cada módulo.

Resistencia a las cargas de viento y nieve especificadas.

El material estructural de los módulos se corresponde con el material especificado por el fabricante.

El proceso de mecanizado para la sujeción del panel y los agujeros de fijación ha sido llevado a cabo de forma tal que no afecte sensiblemente a las propiedades del material estructural.

4.2. Microcontrolador.

Descripción

El microcontrolador utilizado en el presente proyecto es una placa Arduino UNO R3 con un chip ATmega16U2

El conexionado deberá realizarse tal y como se indica en el presente proyecto, puesto que sigue las instrucciones proporcionadas por el fabricante.

4. Pliego de condiciones

Especificaciones.

En caso de utilizar una versión diferente de microcontrolador, se deberá estudiar en qué afecta esto al programa para realizarle las modificaciones necesarias al programa.

El microcontrolador deberá cumplir las características descritas por el fabricante, con especial atención en:

Las tolerancias aplicadas en los pines de entradas y salidas tanto analógicas como digitales abarcan el rango real de valores proporcionados por la placa.

Las variables y funciones que lleva implementadas de serie el microcontrolador y sus variantes para este modelo funcionan de forma acorde a lo dispuesto en el manual del controlador. En cuanto a precisión, exactitud y velocidad en el cálculo, consistencia en el bucle rutinario y fiabilidad.

Todas las conexiones deberán hacerse de acuerdo a lo dispuesto en las especificaciones que aparecen en el datasheet del fabricante.

4.3. Estructura y cimentación.

Descripción

Todos los elementos estructurales han sido fabricados con acero S275 y de forma conforme a la norma UNE EN 10027-1

La cimentación ha sido realizada de acuerdo a lo dispuesto en el presente proyecto, según el Documento EHE 98 CTE para el cálculo de la zapata y el encofrado necesario para el anclaje del seguidor.

Especificaciones

Las uniones articuladas y soldadas entre elementos deberán ser realizadas según el documento del Código Técnico de Edificación CTE DB SE-A y atendiendo al Eurocódigo 3.

5. Presupuesto

5.1. Parte estructural.

Los valores incluyen en precio de instalación.

Barras				
Perfil	Cantidad	Peso (kg)	Precio (€/kg)	Precio total (€)
IPE 100	2	25,92	1,6	82,944
T 20X20	2	2,09	1,6	6,688
T 30X20	2	1,75	1,6	5,6
T 120X120	1	35,7	1,6	57,12
PL 120X8	2	2,49	1,6	7,968
CHS 200	1	40,52	1,6	64,832
UL 65X42	4	12,81	2,35	120,414

Elemento	Cantidad	Área (m ²)	Precio(€/m ²)	Precio total (€)
Plancha 2,2x1,1	1	2,42	6,99	16,9158

Anclaje				
Elemento	Cantidad	Peso (kg)	Precio (€/kg)	Precio total(€)
Placa anclaje	1	13,46	2,24	30,15

Cimentación				
Elemento	Cantidad	m ³	Precio (€/m ³)	Precio total(€)
Hormigonado	1	2,53	97,45	246,5485

		kg	Precio(€/kg)	
Armadura	1	75,59	1,07	80,8813

		m ²	Precio(€/m ²)	
Encofrado	1	4,5	15,55	69,975

397,4048

Mecanismo y transmisión			
Artículo	Cantidad	Precio (€/unidad)	Precio total (€)
Husillo + tuerca	2	64,78	259,12
Piñón engranaje	1	89,38	89,38
Rodamiento	1	287,5	287,5

Total:	1296,48€
--------	----------

5. Presupuesto

5.2. Parte electrónica

Artículo	Cantidad	(€/unidad)	Precio total (€)
Aduino UNO	1	10	10
Motor 3,1Nm Nema23 M23-3.1-1D10-D2-A	2	65	130
Motor 4,2Nm Nema34-4.2-1SB12-E2-A	1	80	80
Final carrera	2	0,85	1,7
Resistencia 10k	2	0,04	0,08
Integrado L293D	2	3,4	6,8
Placa virgen para imprimir circuitos	2	1,5	3
Caja empalmes 16,5 x 11,5 x 7 cm	1	3,8	3,8
Pack 20 Cables 2.54mm + conectores	1	1	1
Regleta de conexión 6 mm 41A Blanco	1	1,85	1,85
30m Cable para steppers UL1007 AWG28	1	7,96	7,96

Total:	246,19
--------	--------

5.3.Presupuesto total

Parte Mecánica	1296,48
Parte Electrónica	246,19

Subtotal:	1542,67
-----------	---------

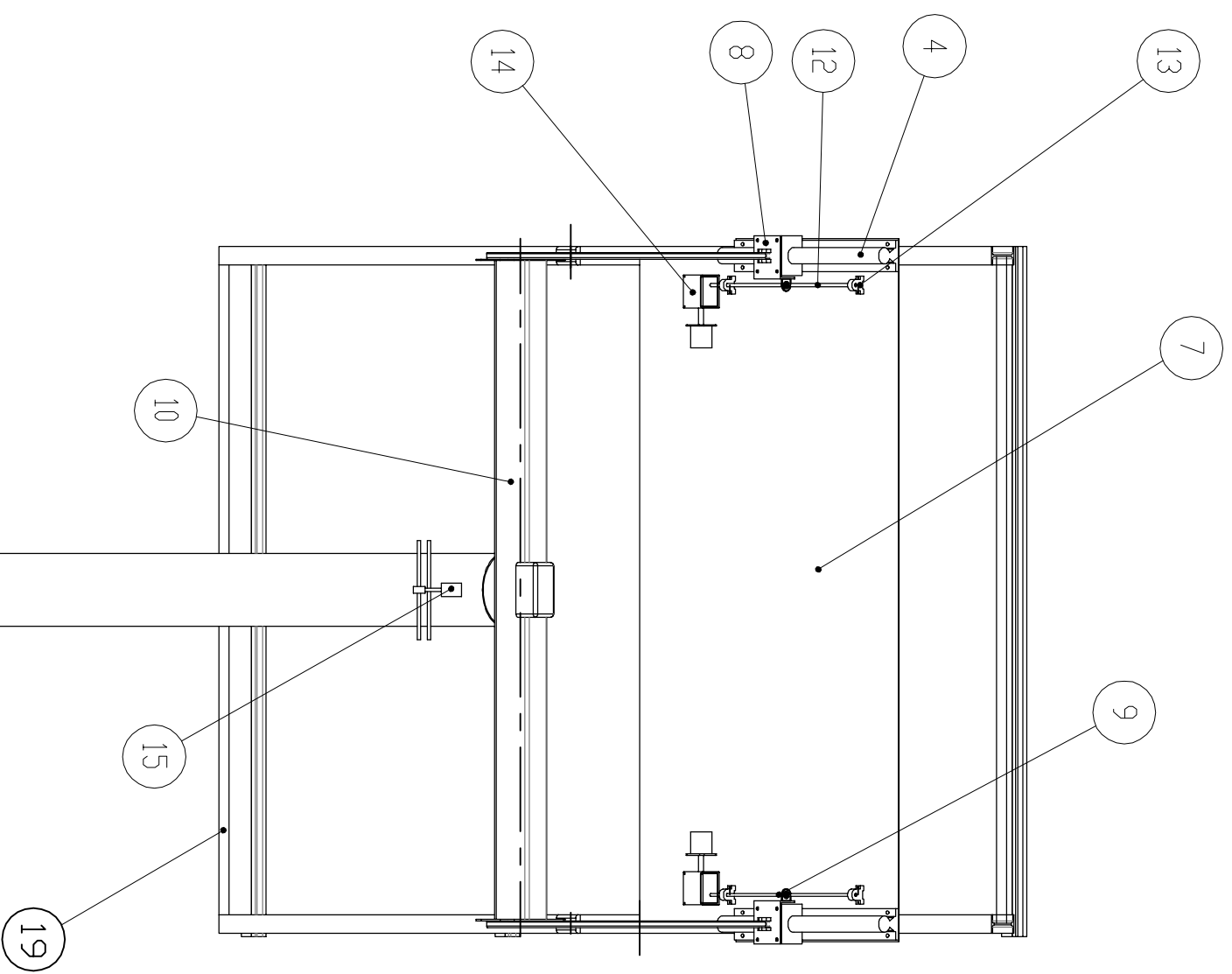
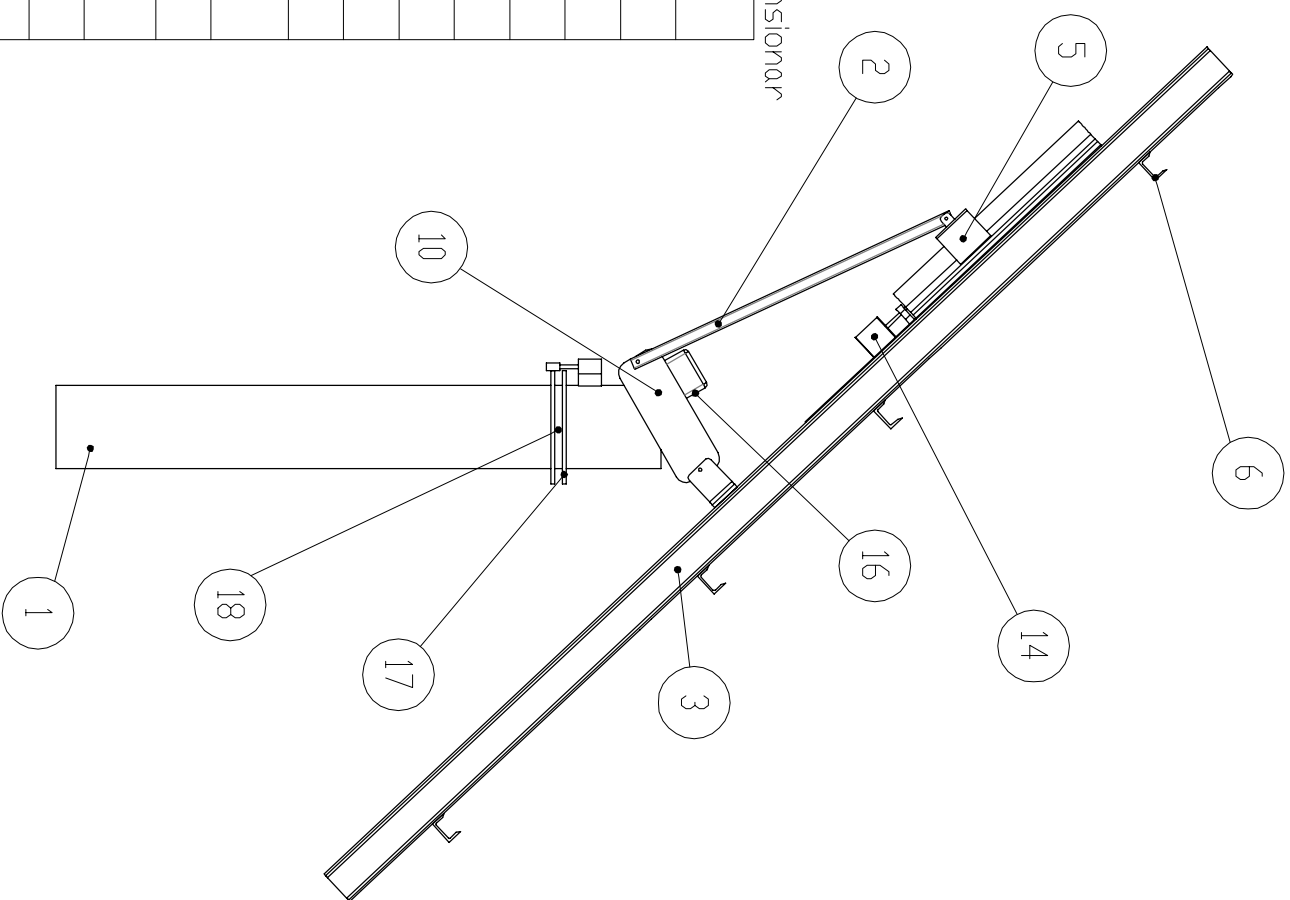
+20%* dimensionado uniones	259,3
+10% imprevistos	154,28
+10% beneficio industrial	154,28

Total:	2110,53
--------	---------

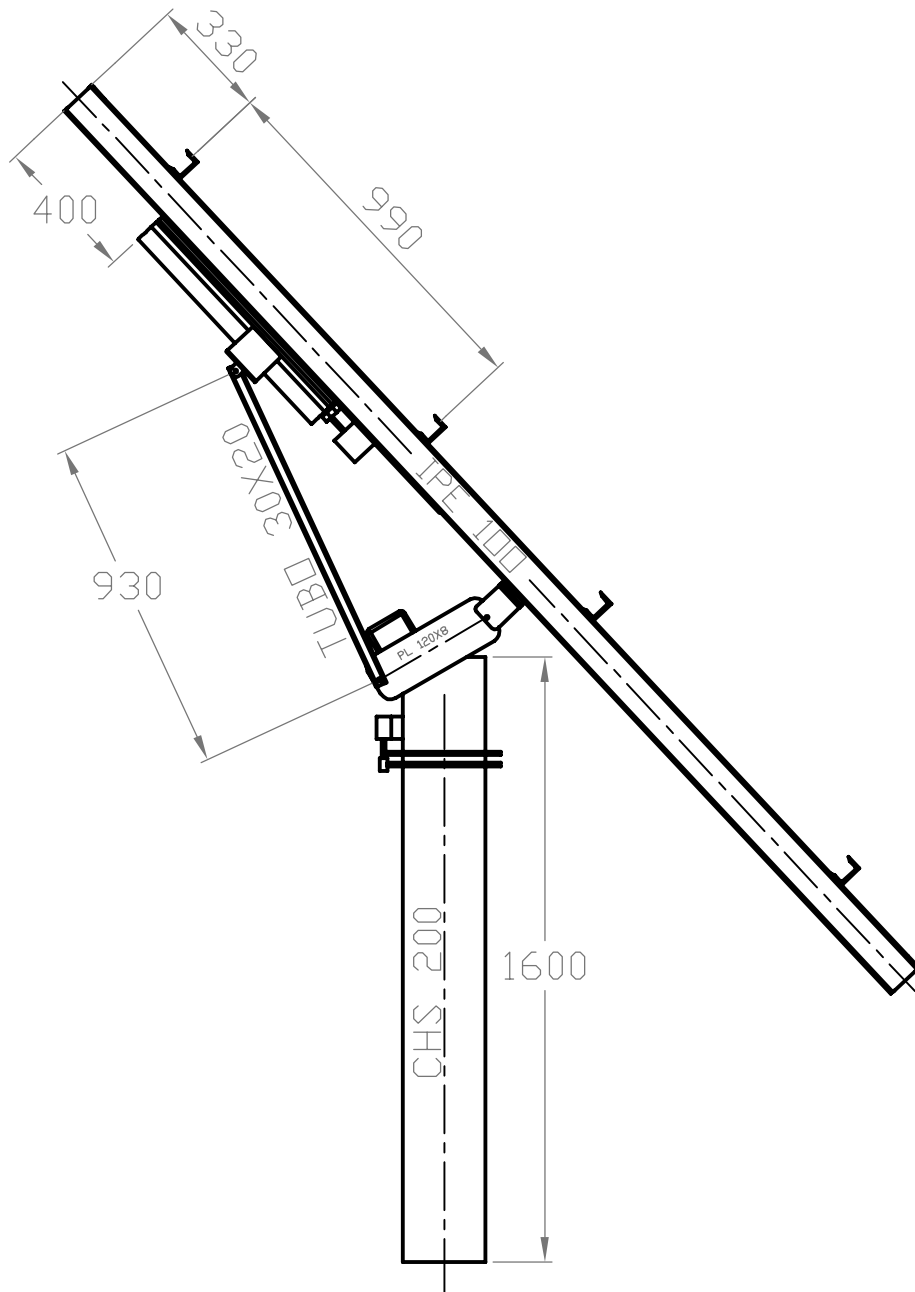
**Se aplica únicamente al coste total de la parte mecánica*

*Estos elementos están pendientes de dimensionar

MARCA	DESCRIPCION	CANTIDAD
1	Mástil CHS200	1
2	Barra móvil 30x20	2
3	Larguero IPE100	2
4	Guía 600mm	2
5	Seguidor guía	2
6	Correa UL65x42	4
7	Placa union trasera	1
8	Unión tuercas y deslizador*	2
9	Tuerca del sinfin	2
10	Tubo transversal 120x120 + PL120x8	1
11	Articulacion IPE*	2
12	Tornillo sinfin 600mm	2
13	Sujecion y cojinete*	4
14	Stepper con reductora (elevación)	2
15	Stepper con reductora (rotación)	1
16	Caja empalmes	1
17	Junta mástil*	1
18	Rodamiento axial*	1
19	Montante 20x20	2

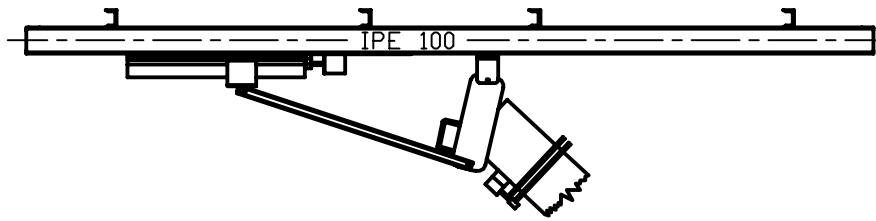


Observaciones:		Título:		Plano nº:	
ESCALA: 1/20		Unidad dim. mm		1	
				Plano de conjunto	
		ETSCE		Dirigido por: Sergio Balaguer	
				Fecha: 08/11/2016	
				HDJA 1 DE 12	

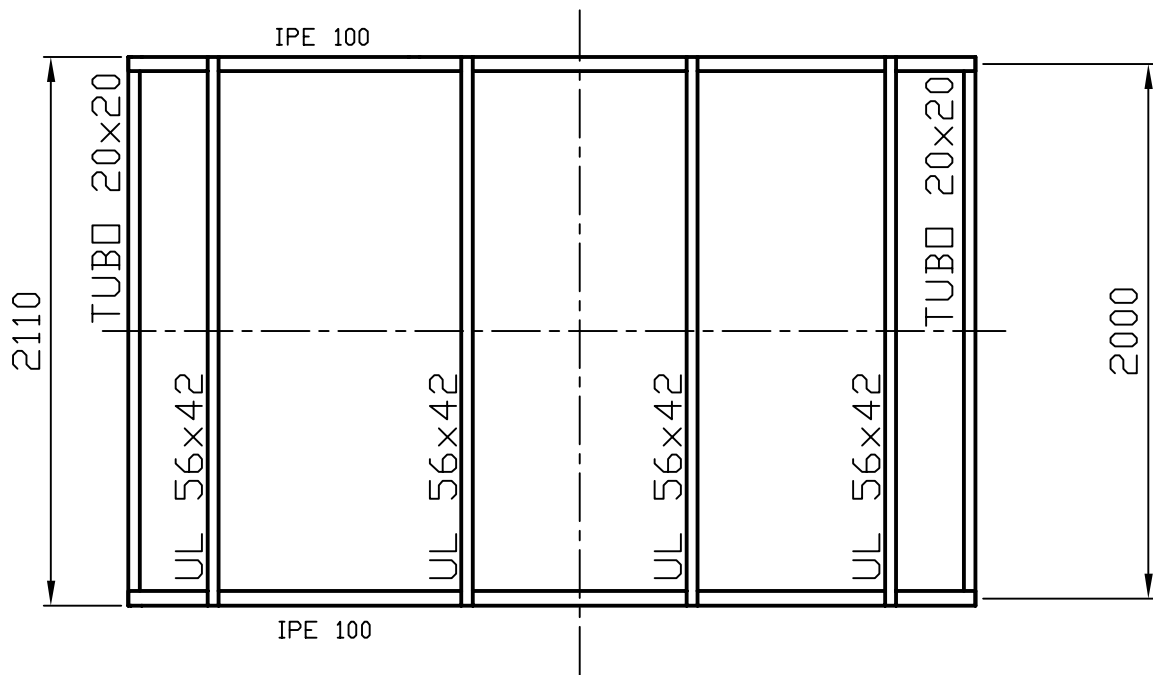


Observaciones:		Título: Dimensiones principales del seguidor A4		Plano nº : 2
				HOJA 2 DE 12
ESCALA: 1:20	Unidad dim. mm 	Escuela Superior de Tecnología	Dirigido por: Sergio Balaguer	Fecha: 08/11/2016

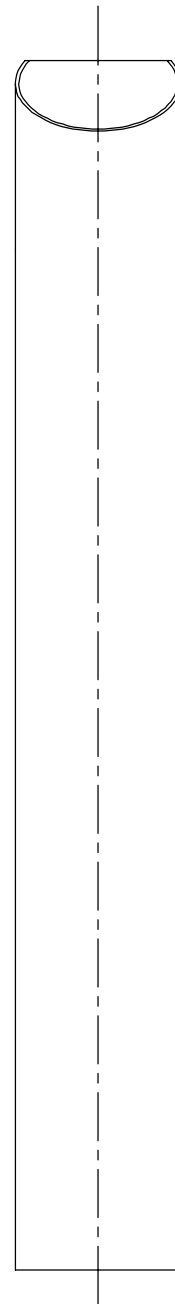
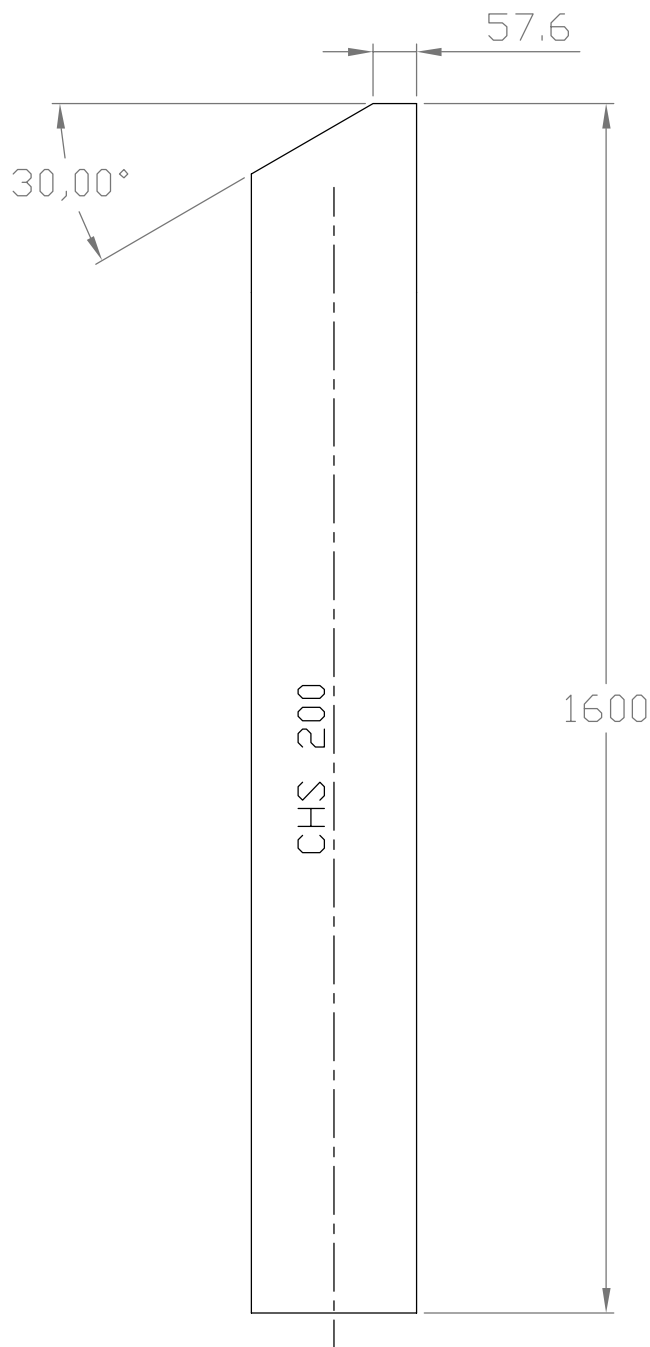
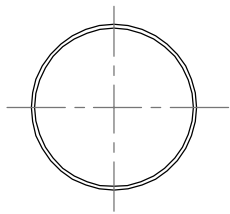
A ↓



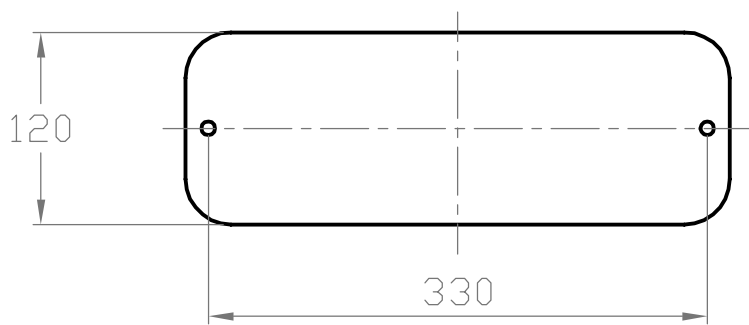
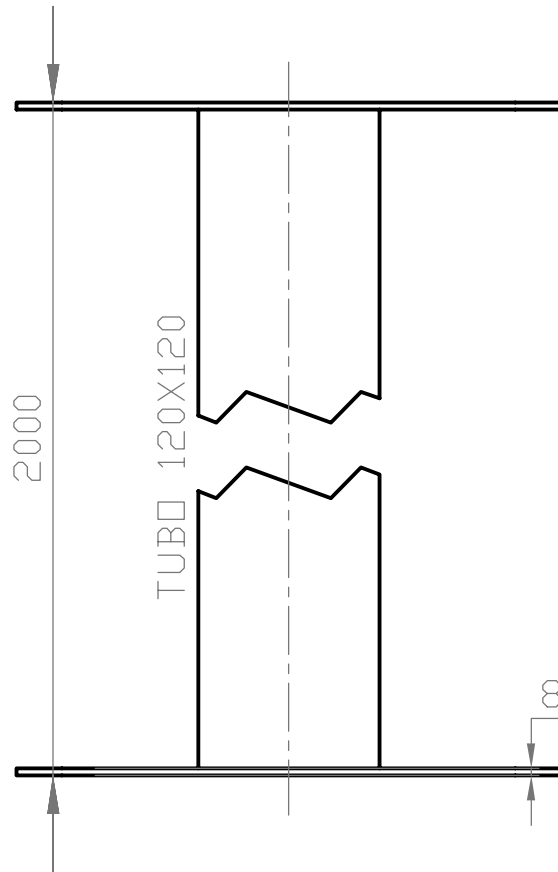
Vista A



Observaciones:	Título: Vista normal del bastidor A4	Plano nº: 3
		HOJA 3 DE 12
ESCALA: 1:25	Unidad dim. mm 	Dirigido por: Sergio Balaguer
Escuela Superior de Tecnología UNIVERSITAT JAUME I		Fecha: 08/11/2016

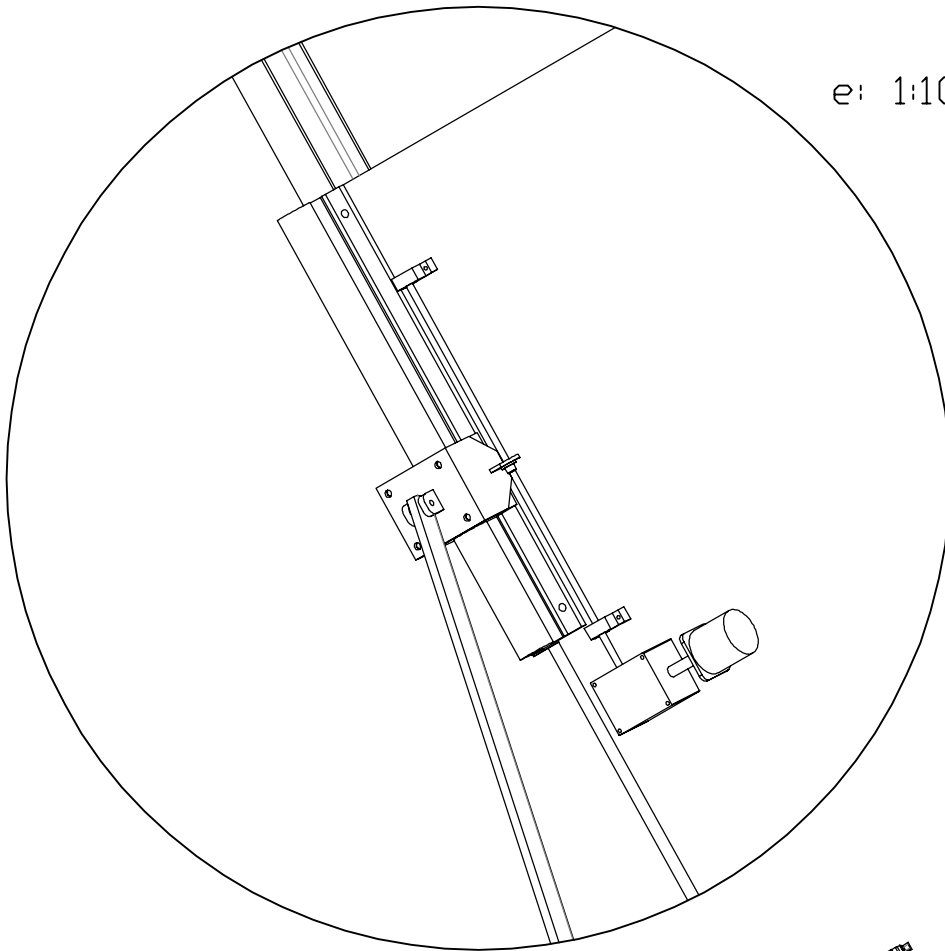


Observaciones:	Título: Pieza para el mástil CHS 200 (Sin la junta ni los rodamientos)	Plano nº : 4		
		HOJA 4 DE 12		
ESCALA: 1:10	Unidad dim. mm	 Escuela Superior de Tecnología	Dirigido por: Sergio Balaguer	Fecha: 08/11/2016
				

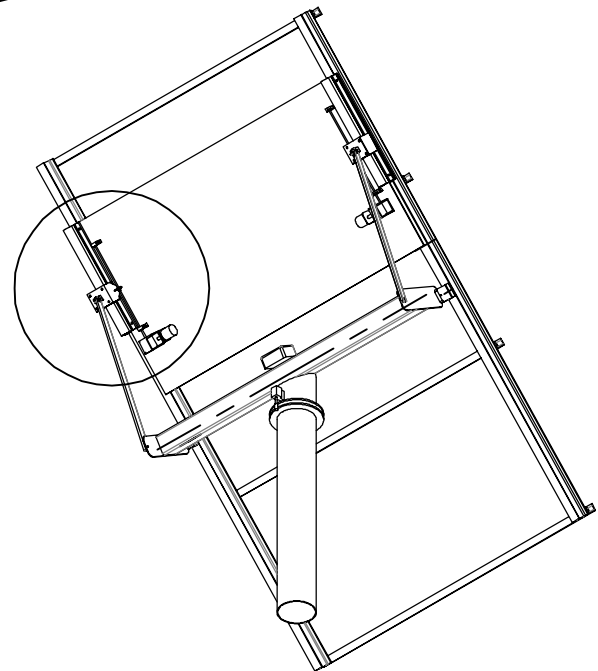



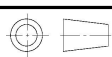
Nota: Se han acotado únicamente las magnitudes necesarias para el montaje. El resto de parámetros deberán ser determinados en el momento del dimensionado de las uniones

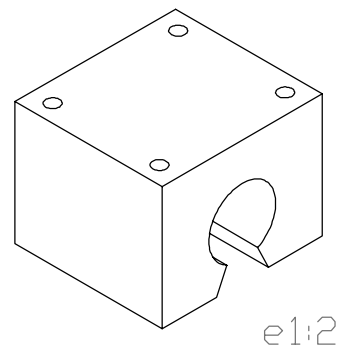
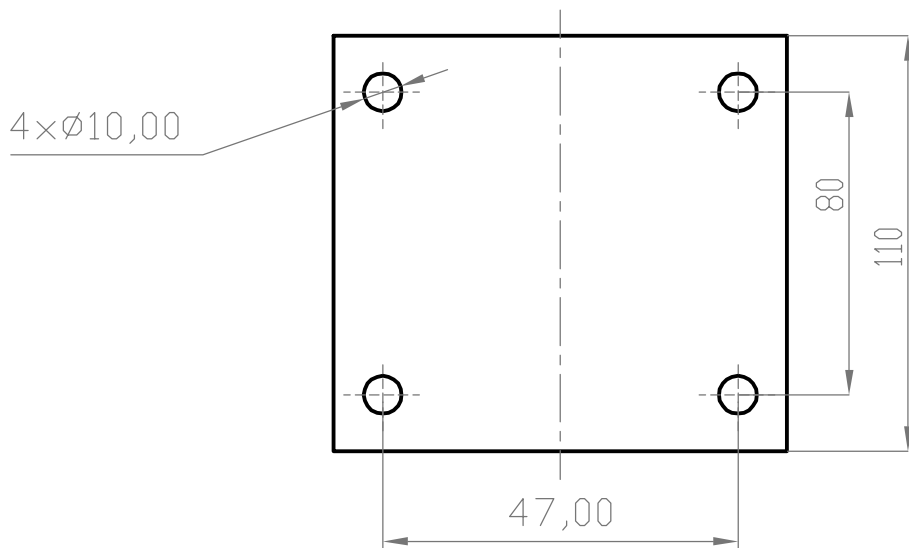
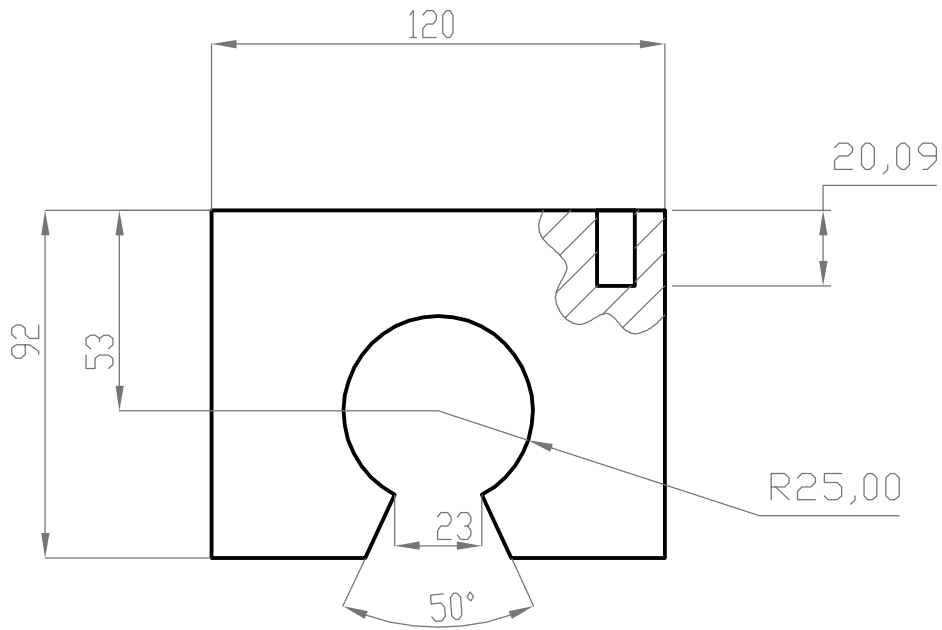
Observaciones:	Título: Tubo transversal Tubo 120x120 + 2xPL120x8	A4	Plano nº : 5
			HOJA 5 DE 12
ESCALA: 1:5	Unidad dim. mm 	Escuela Superior de Tecnología UNIVERSITAT JAUME I	Dirigido por: Sergio Balaguer
			Fecha: 08/11/2016



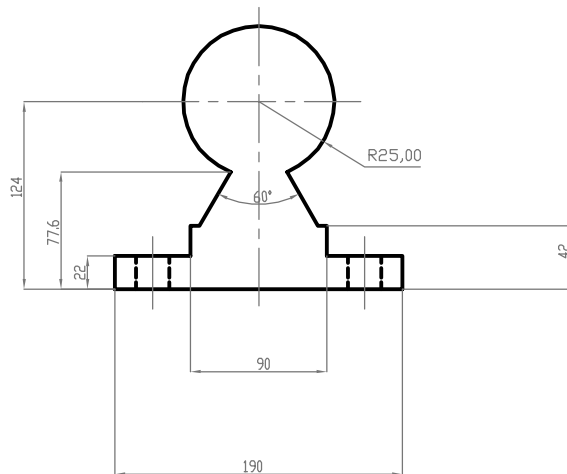
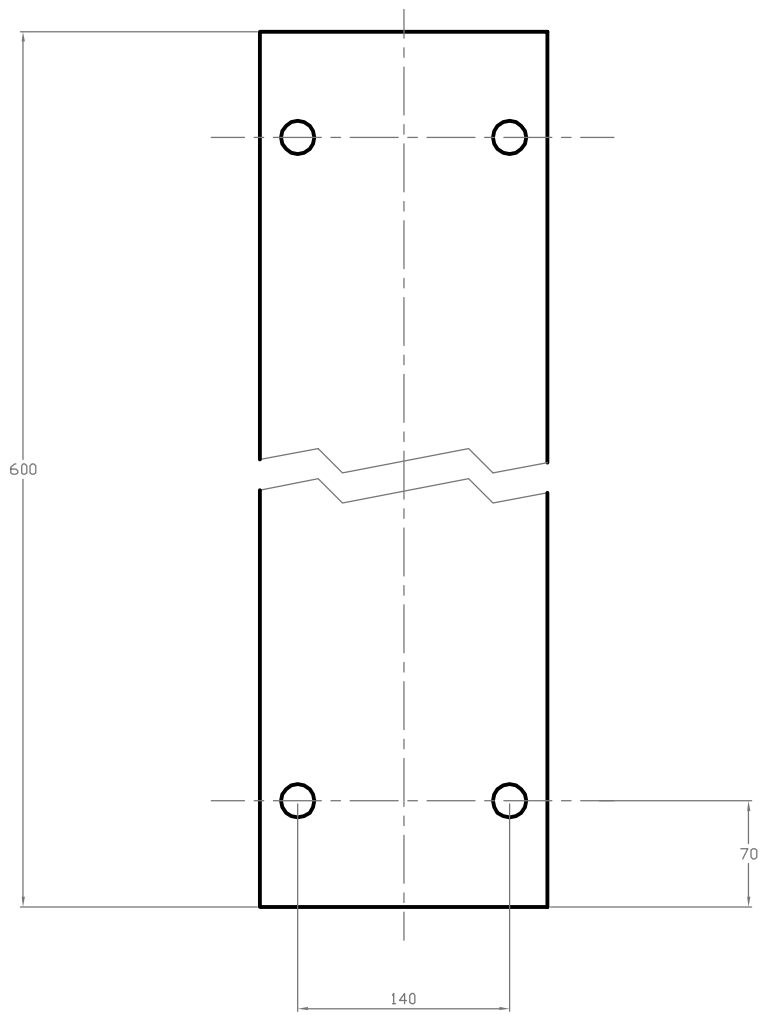
e: 1:10



Observaciones:	Título: Detalle del sistema de elevación A4	Plano nº: 6
		HOJA 6 DE 12
Unidad dim. mm	 Escuela Superior de Tecnología	Dirigido por: Sergio Balaguer
		Fecha: 08/11/2016



Observaciones:		Título: Deslizadera (sistema de guiado) A4		Plano nº: 7
				HOJA 7 DE 12
ESCALA: 1:1	Unidad dim. mm 	Escuela Superior de Tecnología	Dirigido por: Sergio Balaguer	Fecha: 08/11/2016



Observaciones:

Título:

Guia 600mm

Plano nº :

8

A4

HOJA 8 DE 12

ESCALA:
1:2

Unidad dim. mm

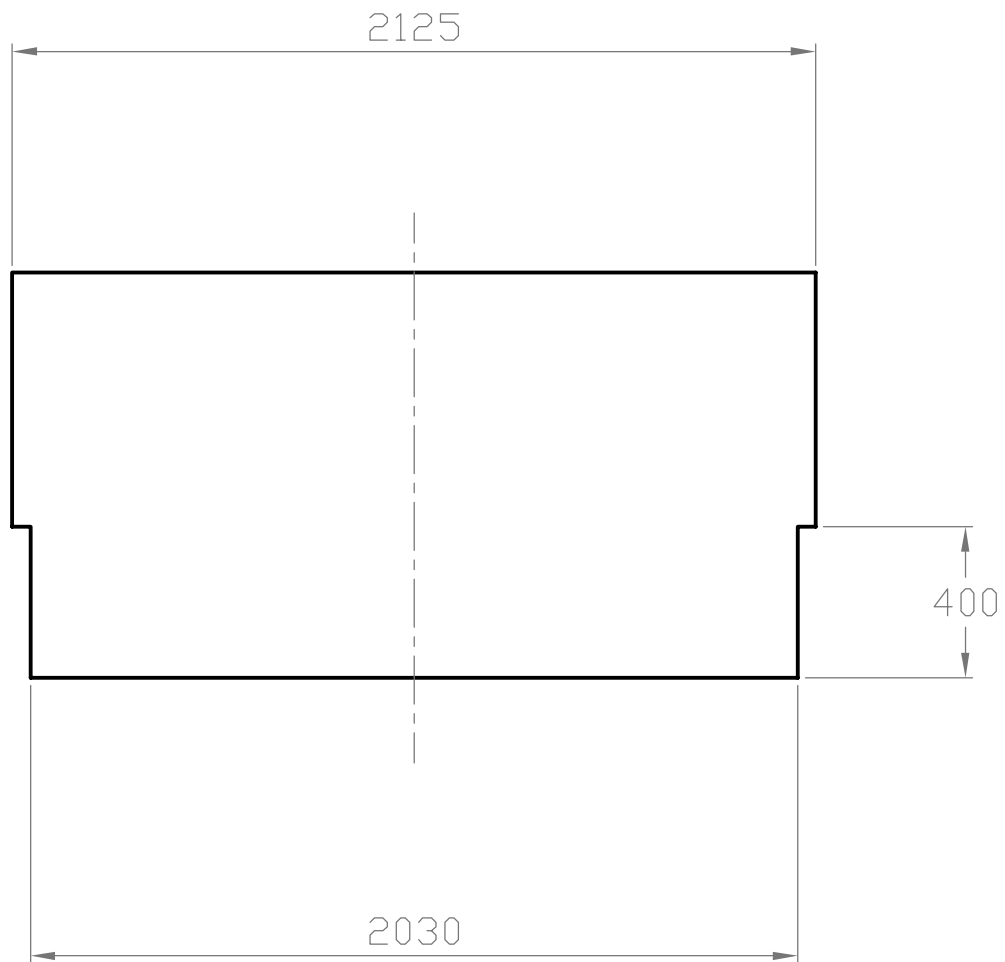


Escuela
Superior de
Tecnología
UNIVERSITAT
JAUME I

Dirigido por:

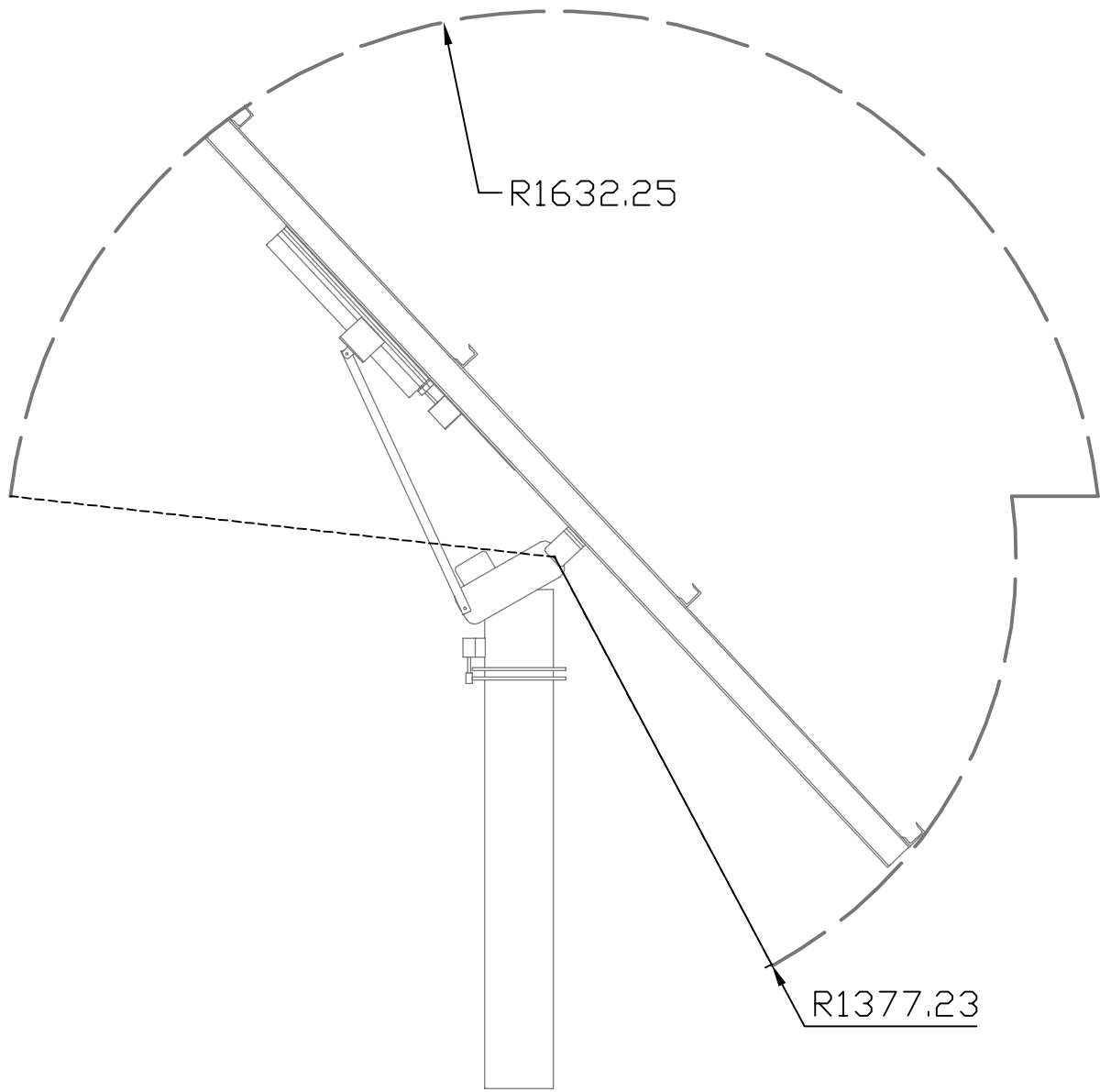
Sergio Balaguer

Fecha: 08/11/2016



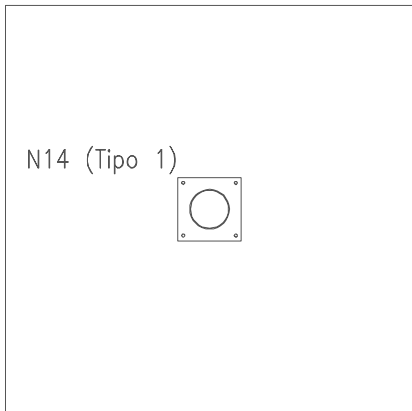
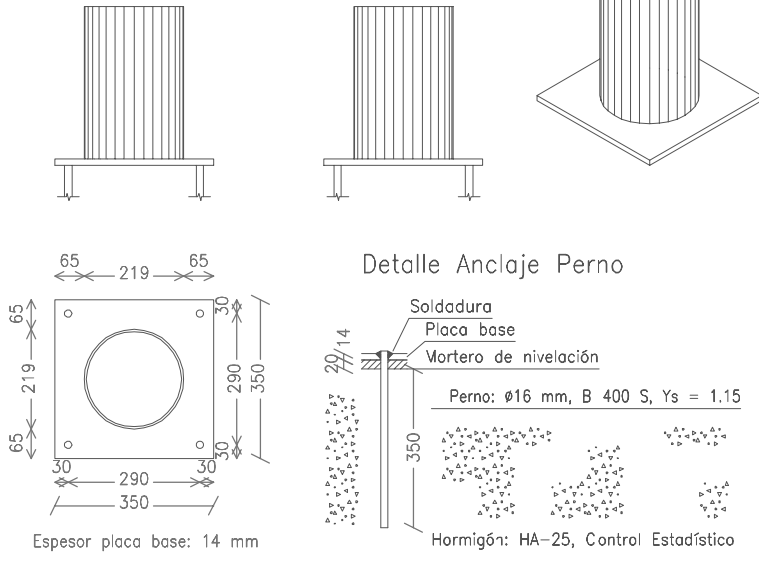
Espesor: 6mm

Observaciones:		Título: Placa uniones trasera A4		Plano nº : 9
				HOJA 9 DE 12
ESCALA: 1:20	Unidad dim. mm 	 Escuela Superior de Tecnología UNIVERSITAT JAUME I	Dirigido por: Sergio Balaguer	Fecha: 08/11/2016



Observaciones:	Título: Volumen de trabajo	A4	Plano nº : 10
			HOJA 10 DE 12
ESCALA: 1:20	Unidad dim. mm 	Escuela Superior de Tecnología UNIVERSITAT JAUME I	Dirigido por: Sergio Balaguer
			Fecha: 08/11/2016

Tipo 1
 Dimensiones Placa = 350x350x14 mm (S275)
 Pernos = 4Ø16 mm, B 400 S, Ys = 1.15
 Ref. pilares : N14
 Escala 1 : 20



225 x 225 x 50

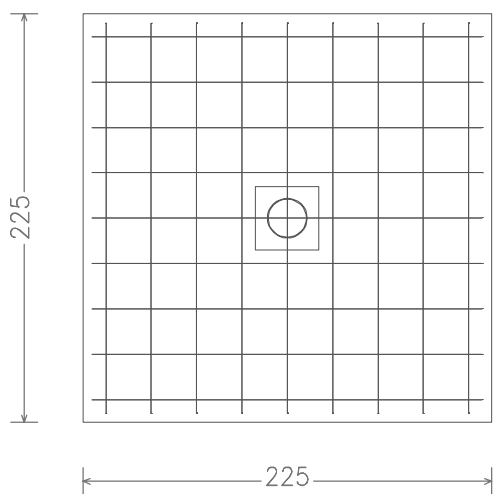
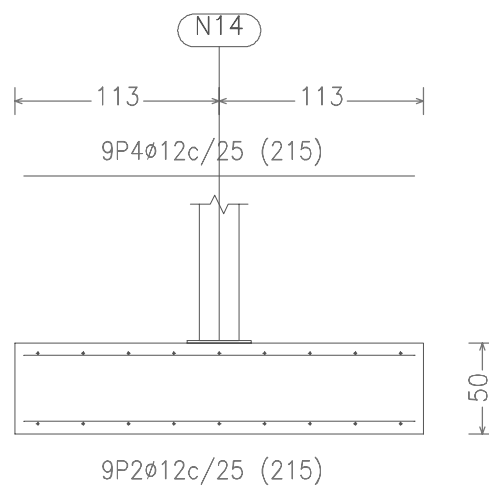
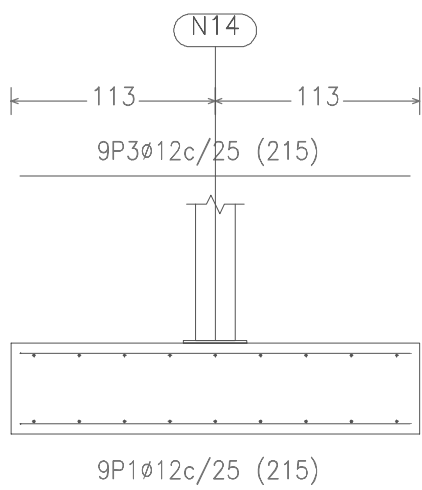
Resumen Acero Elemento y Placa de anclaje	Long. total (m)	Peso+10% (kg)
B 500 S, CN	77.4	76

Cuadro de arranques		
Referencias	Pernos de Placas de Anclaje	Dimensión de Placas de Anclaje
N14	4 Pernos Ø 16	Placa base (350x350x14)

Elemento	Pos.	Díam.	No.	Long. (cm)	Total (cm)	B 500 S, CN (kg)
N14	1	Ø12	9	215	1935	17.2
	2	Ø12	9	215	1935	17.2
	3	Ø12	9	215	1935	17.2
	4	Ø12	9	215	1935	17.2
Total+10%:						75.7
Ø12:						75.7
Total:						75.7

Observaciones:	Título: Placa de anclaje	A4	Plano nº : 11
			HOJA 11 DE 12
ESCALA: 1:50	Unidad dim. cm	Dirigido por: Sergio Balaguer	Fecha: 08/11/2016

N14



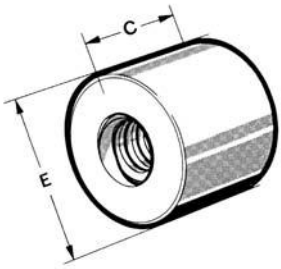
Observaciones:	Título: Zapata y mallazo	Plano nº: 12
ESCALA: 1:50	Unidad dim. cm	Dirigido por: Sergio Balaguer
	<p>Escuela Superior de Tecnología UNIVERSITAT JAUME·I</p>	Fecha: 08/11/2016

Tuercas Trapezoidales

Tuerca larga cilíndrica LRM

- Para aplicaciones de velocidades lentas y medias, con ciclo de trabajo menos de 20%
- Puede ser utilizada en combinaciones con husillos de inox. resistentes a la corrosión

- Tratamiento adicional: la rosca sirve como referencia para un proceso de fabricación y montaje preciso
- Material: 2.1090 (G-CuSn 7 ZnPb (Rg7))



Tipo	E (mm)	C (mm)	Peso (kg)	(mm ²)	Tipo	E (mm)	c (mm)	Peso (kg)	(mm ²)
LRM Tr 10x2	22	20	0.056	200	LRM Tr 26x5	50	48	0.58	1240
LRM Tr 10x3	22	20	0.056	190	LRM Tr 28x5	60	60	1.2	1680
LRM Tr 12x3	26	24	0.092	280	LRM Tr 30x6	60	60	1.2	1780
LRM Tr 12x6 P3 ²⁾	26	24	0.092	280	LRM Tr 30x12 P6	60	60	1.2	1780
LRM Tr 14x3	30	28	0.14	380	LRM Tr 32x6	60	60	1.2	1910
LRM Tr 14x4	30	28	0.14	370	LRM Tr 36x6	75	72	2.2	2610
LRM Tr 16x4	36	32	0.25	490	LRM Tr 40x7	80	80	2.8	3210
LRM Tr 16x8 P4 ²⁾	36	32	0.25	490	LRM Tr 40x14 P7	80	80	2.8	3210
LRM Tr 18x4	40	36	0.34	630	LRM Tr 44x7	80	80	2.6	3560
LRM Tr 20x4	45	40	0.48	790	LRM Tr 48x8	90	100	4.3	4840
LRM Tr 20x8 P4 ²⁾	45	40	0.45	790	LRM Tr 50x8	90	100	4.2	5060
LRM Tr 22x5	45	40	0.46	850	LRM Tr 60x9	100	120	5.7	7320
LRM Tr 22x24 P4 ¹⁾	45	40	0.46	880	LRM Tr 70x10	110	140	7.6	10000
LRM Tr 24x5	50	48	0.69	1130	LRM Tr 80x10	120	160	9.7	13200
LRM Tr 24x10 P5 ²⁾	50	48	0.65	1130					

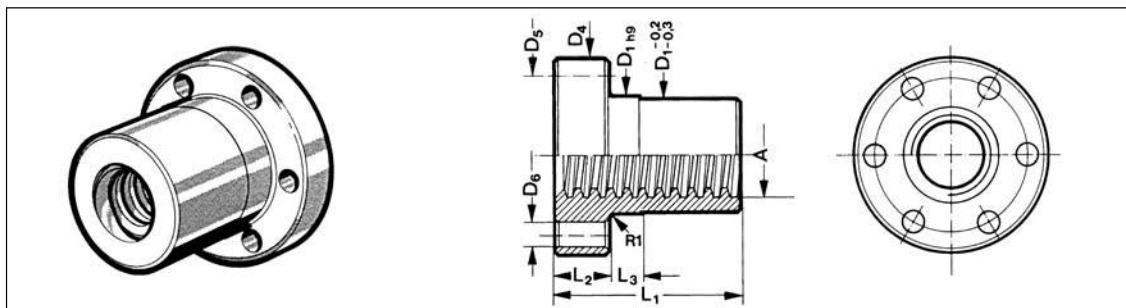
¹⁾ Rosca especial; diámetro 21,5

²⁾ P3: Paso 3, 2 entradas

Tuercas completas de bronce - EFM

- Para transmisiones de movimiento en operación continua, con unas características de uso particularmente buenas.
- Apropriadadas para ser utilizadas como tuerca de seguridad
- Pueden ser utilizadas en combinación con husillos inox. resistentes a la corrosión.

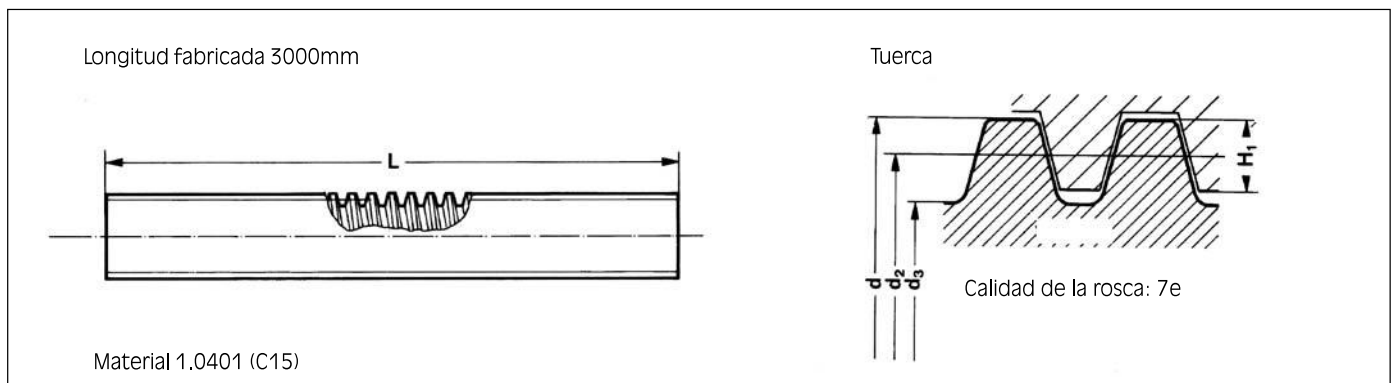
- Las tuercas EFM pueden ser montadas con los adaptadores KON y KAR.
- Material: 2.1090(G-CuSn(Rg7))



Tipo	Dimensiones (mm)							(kg)	(mm ²)
	D ₁	D ₄	D ₅	6xD ₆	L ₁	L ₂	L ₃		
EFM Tr 16x4	28	48	38	6	44	12	8	0.25	670
EFM Tr 18x4	28	48	38	6	44	12	8	0.25	770
EFM Tr 20x4	32	55	45	7	44	12	8	0.30	870
EFM Tr 24x5	32	55	45	7	44	12	8	0.30	1040
EFM Tr 30x6	38	62	50	7	46	14	8	0.40	1370
EFM Tr 36x6	45	70	58	7	59	16	10	0.60	2140
EFM Tr 40x7	63	95	78	9	73	16	10	1.7	2930
EFM Tr 50x8	72	110	90	11	97	18	10	2.6	4900
EFM Tr 60x9	85	125	105	11	99	20	10	3.7	6040

Husillos trapezoidales

Husillos trapezoidales de precisión laminados RPTS



Tipo ¹⁾	Dimensiones					Precisión $\mu\text{mm}/$ 300 mm	Rectitud $\mu\text{mm}/$ 300 mm	2)	3)	Peso (kg/m)	Momento de inercia polar lcm ⁴¹	Momento de resistencia ⁴⁾ lcm ³¹	Momento de inercia (kgm ² /m)
	d	d _{2 min}	d _{2 max}	d ₃ ⁷⁾	H ₁								
RPTS Tr 10x2	10	8.739	8.929	6.89	1	300	0.5	4°2'	0.40	0.500	0.011	0.032	0.51·10 ⁻⁵
RPTS Tr 10x3	10	8.191	8.415	5.84	1.5	300	0.5	6°24'	0.51	0.466	0.0057	0.020	0.40·10 ⁻⁵
RPTS Tr 12x3	12	10.191	10.415	7.84	1.5	300	0.5	5°11'	0.46	0.746	0.019	0.047	1.03·10 ⁻⁵
RPTS Tr 12x6 P3	12	10.165	10.415	7.84	1.5	300	0.5	10°18'	0.62	0.746	0.019	0.047	1.03·10 ⁻⁵
RPTS Tr 14x3	14	12.191	12.415	9.84	1.5	300	0.5	4°22'	0.42	1.04	0.046	0.094	2.04·15 ⁻⁵
RPTS Tr 14x4	14	11.640	11.905	8.80	2	300	0.5	6°3'	0.50	0.888	0.029	0.067	1.60·10 ⁻⁵
RPTS Tr 16x4	16	13.640	13.905	10.80	2	50	0.1	5°11'	0.46	1.21	0.067	0.124	2.96·10 ⁻⁵
RPTS Tr 16x8 P4	16	13.608	13.905	10.80	2	300	0.3	10°18'	0.62	1.20	0.067	0.124	2.96·10 ⁻⁵
RPTS Tr 18x4	18	15.640	15.905	12.80	2	50	0.1	4°32'	0.43	1.58	0.132	0.206	5.05·10 ⁻⁵
RPTS Tr 20x4 ⁶⁾	20	17.640	17.905	14.8	2	50	0.1	4°2'	0.40	2.00	0.236	0.318	8.10·10 ⁻⁵
RPTS Tr 20x8 P4	20	17.608	17.905	14.8	2	200	0.2	8°3'	0.57	2.00	0.236	0.318	8.10·10 ⁻⁵
RPTS Tr 20x16 P4	20	17.608	17.905	14.8	2	200	0.2	15°47'	0.71	2.00	0.236	0.318	8.10·10 ⁻⁵
RPTS Tr 22x5	22	19.114	19.394	15.50	2.5	50	0.1	4°39'	0.43	2.23	0.283	0.366	1.00·10 ⁻⁴
RPTS Tr 22x24 P4S5 ⁵⁾	22	19.140	19.505	16.50	2.5	200	0.2	21°34'	0.75	2.23	0.364	0.411	1.00·10 ⁻⁴
RPTS Tr 24x5	24	21.094	21.394	17.50	2.5	50	0.1	4°14'	0.41	2.72	0.460	0.526	1.50·10 ⁻⁴
RPTS Tr 24x10P5	24	21.058	21.394	17.50	2.5	200	0.2	8°25'	0.58	2.72	0.460	0.526	1.50·10 ⁻⁴
RPTS Tr 26x5	26	23.094	23.394	19.50	2.5	50	0.1	3°52'	0.39	3.26	0.710	0.728	2.0·10 ⁻⁴
RPTS Tr 28x5	28	25.094	25.394	21.50	2.5	50	0.1	3°34'	0.37	3.85	1.05	0.976	3.0·10 ⁻⁴
RPTS Tr 30x6	30	26.547	26.882	21.90	3	50	0.1	4°2'	0.40	4.50	1.13	1.03	4.0·10 ⁻⁴
RPTS Tr 30x12 P6	30	26.507	26.882	21.90	3	200	0.2	8°3'	0.57	4.50	1.13	1.03	4.0·10 ⁻⁴
RPTS Tr 32x6	32	28.547	28.882	23.90	3	50	0.1	3°46'	0.38	5.18	1.60	1.34	5.0·10 ⁻⁴
RPTS Tr 36x6	36	32.547	32.882	27.90	3	50	0.1	3°18'	0.35	6.71	2.97	2.13	9.0·10 ⁻⁴
RPTS Tr 40x7	40	36.020	36.375	30.50	3.5	50	0.1	3°29'	0.37	8.00	4.25	2.79	1.3·10⁻³
RPTS Tr 40x14 P7	40	35.978	36.375	30.50	3.5	200	0.2	6°57'	0.53	8.00	4.25	2.79	1.3·10 ⁻³
RPTS Tr 44x7	44	40.020	40.275	34.50	3.5	50	0.1	3°8'	0.34	9.87	6.95	4.03	2.0·10 ⁻³
RPTS Tr 48x8	48	43.468	43.868	37.80	4	100	0.1	3°18'	0.35	12.0	10.0	5.30	2.9·10 ⁻³
RPTS Tr 50x8	50	45.468	45.868	39.30	4	100	0.1	3°10'	0.34	13.1	11.7	5.96	3.4·10 ⁻³
RPTS Tr 60x9	60	54.935	55.360	48.15	4.5	200	0.3	2°57'	0.33	18.0	26.4	11.0	6.9·10 ⁻³
RPTS Tr 70x10	70	64.425	64.850	57.00	5	200	0.3	2°48'	0.32	26.0	51.8	18.2	1.4·10 ⁻⁴
RPTS Tr 80x10	80	74.425	74.850	67.00	5	200	0.3	2°25'	0.29	34.7	98.9	29.5	2.4·10 ⁻³

¹⁾ Tr 20x4 significa: rosca trapezoidal con un diámetro exterior d=20mm, de una entrada, con paso de 4mm con rosca derecha o izquierda

Tr 20x8 P4 significa: rosca trapezoidal con un diámetro exterior d=20mm, de dos entradas, con paso de 8mm, con sólo rosca derecha

²⁾ Ángulo de inclinación de la rosca: ver fórmula pág.51

³⁾ El coeficiente teórico para convertir el movimiento rotativo a movimiento lineal con coeficiente de fricción $\mu = 0.1$. Eficacia para otros coeficientes de fricción; ver fórmula pág.51

⁴⁾ El momento de inercia polar es el doble que el momento de inercia

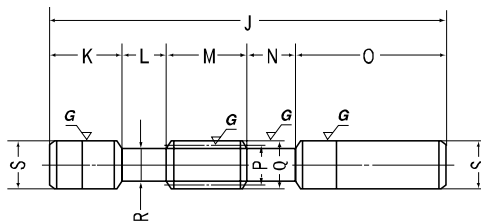
⁵⁾ Perfil especial

⁶⁾ Hasta una longitud de 6000mm. Se realizan cortes según la medida deseada por el cliente

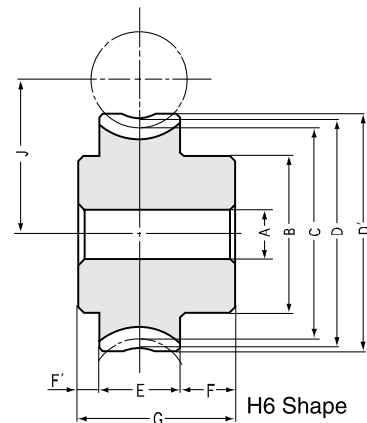
⁷⁾ El diámetro de fondo es según norma DIN 103



KWG Ground Worm Shafts, AGF Worm Wheels Axial Modules 5~6



W6 Shape



Module 5 Ground Worm Shafts

Catalog No.	Axial module <i>m</i>	Number of starts	Lead angle	Hand of thread	Total length	Shaft length (L)	Neck length (L)	Face width	Neck length (R)	Shaft length (R)	Pitch dia.	Outside dia.	Neck dia.
					J	K	L	M	N	O	P	Q	R
KWG5-R1	5	1	5°43'	R	400	75	30	90	30	175	50	60	36

Module 5 Worm Wheels

Catalog No.	Transverse module <i>m</i>	Reduction ratio	No. of teeth	Coefficient of profile shift	Number of starts	Hand of thread	Bore	Hub dia.	Pitch dia.	Throat dia.	Outside dia.	Face width	Hub length (R)	Hub length (L)
							AH7	B	C	D	D'	E	F	F'
AGF5-20R1	5	1/20	20	0	1	R	22	75	100	110	115	35	23	12
AGF5-25R1	5	1/25	25	0	1	R	22	75	125	135	140	35	23	12
AGF5-30R1	5	1/30	30	0	1	R	22	75	150	160	165	35	23	12
AGF5-36R1	5	1/36	36	0	1	R	22	90	180	190	195	35	23	12
AGF5-40R1	5	1/40	40	0	1	R	22	110	200	210	215	35	23	12
AGF5-48R1	5	1/48	48	0	1	R	140	—	240	250	255	35	17.5	7.5
AGF5-50R1	5	1/50	50	0	1	R	150	—	250	260	265	35	17.5	7.5
AGF5-60R1	5	1/60	60	0	1	R	200	—	300	310	315	35	17.5	7.5

CAUTION: Before assembling or performing any secondary operations, please carefully read the "Application Hints" on page 296.

Module 6 Ground Worm Shafts

Catalog No.	Axial module <i>m</i>	Number of starts	Lead angle	Hand of thread	Total length	Shaft length (L)	Neck length (L)	Face width	Neck length (R)	Shaft length (R)	Pitch dia.	Outside dia.	Neck dia.
					J	K	L	M	N	O	P	Q	R
KWG6-R1	6	1	5°43'	R	400	60	40	100	40	160	60	72	44

Module 6 Worm Wheels

Catalog No.	Transverse module <i>m</i>	Reduction ratio	No. of teeth	Coefficient of profile shift	Number of starts	Hand of thread	Bore	Hub dia.	Pitch dia.	Throat dia.	Outside dia.	Face width	Hub length (R)	Hub length (L)
							AH7	B	C	D	D'	E	F	F'
AGF6-20R1	6	1/20	20	0	1	R	25	85	120	132	138	40	23	12
AGF6-25R1	6	1/25	25	0	1	R	25	90	150	162	168	40	23	12
AGF6-30R1	6	1/30	30	0	1	R	25	100	180	192	198	40	23	12
AGF6-36R1	6	1/36	36	0	1	R	25	110	216	228	234	40	23	12
AGF6-40R1	6	1/40	40	0	1	R	130	—	240	252	258	40	20	8
AGF6-48R1	6	1/48	48	0	1	R	180	—	288	300	306	40	20	8
AGF6-50R1	6	1/50	50	0	1	R	190	—	300	312	318	40	20	8
AGF6-60R1	6	1/60	60	0	1	R	250	—	360	372	378	40	20	8

CAUTION: Do not take heavy cuts when performing secondary operations on worm shafts to avoid bending them.

AGF5 Allowable Worm Wheel Torques (N.m)

Catalog No.	Worms min ⁻¹						
	30	100	300	600	900	1200	1800
AGF5-20R1	211	172.4	134.2	108.3	95	86.2	72.7
AGF5-25R1	319	261	206	168.2	147.3	133.9	113.7
AGF5-30R1	446	369	291	239	211	190.5	163.5
AGF5-36R1	627	519	414	343	302	274	234
AGF5-40R1	763	632	506	421	371	337	288
AGF5-48R1	1070	886	715	598	530	483	411
AGF5-50R1	1153	956	772	646	574	523	446
AGF5-60R1	1614	1344	1087	913	820	744	639

Worm Gear Pair
KWG · AGF

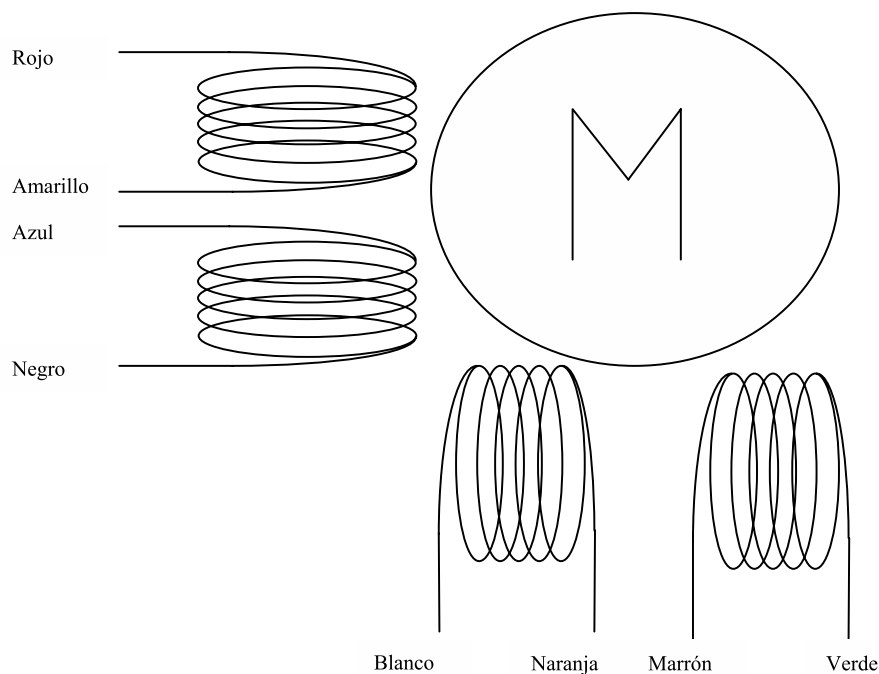
Motor de rotación Nema 23

Tipo	Motor paso a paso híbrido
Normalización	Size 23 series-v3
Paso angular	1,8°
Número de fases	2
Número de hilos	8
ΔT_{max}	80 grados
Clase de aislamiento	B
Resistencia de aislamiento	100 Mohm (500 Vdc)

Tipo	Conexión	Nm	A	Ohm	V	mH	\varnothing	L(mm)	Peso (Kg)
M23-3.1-1D10-D2-A	Paralelo	3.1	4.2	0.65	2.7	3.2	10	88	1.4
	Serie	3.1	2.1	2.6	5.5	12.8	10	88	1.4
	Unipolar	2.2	2.9	1.3	3.8	3.2	10	88	1.4

A = Corriente, Ohm = Resistencia, V = Tensión mH= Inducción propia L=Longitud

Hilos



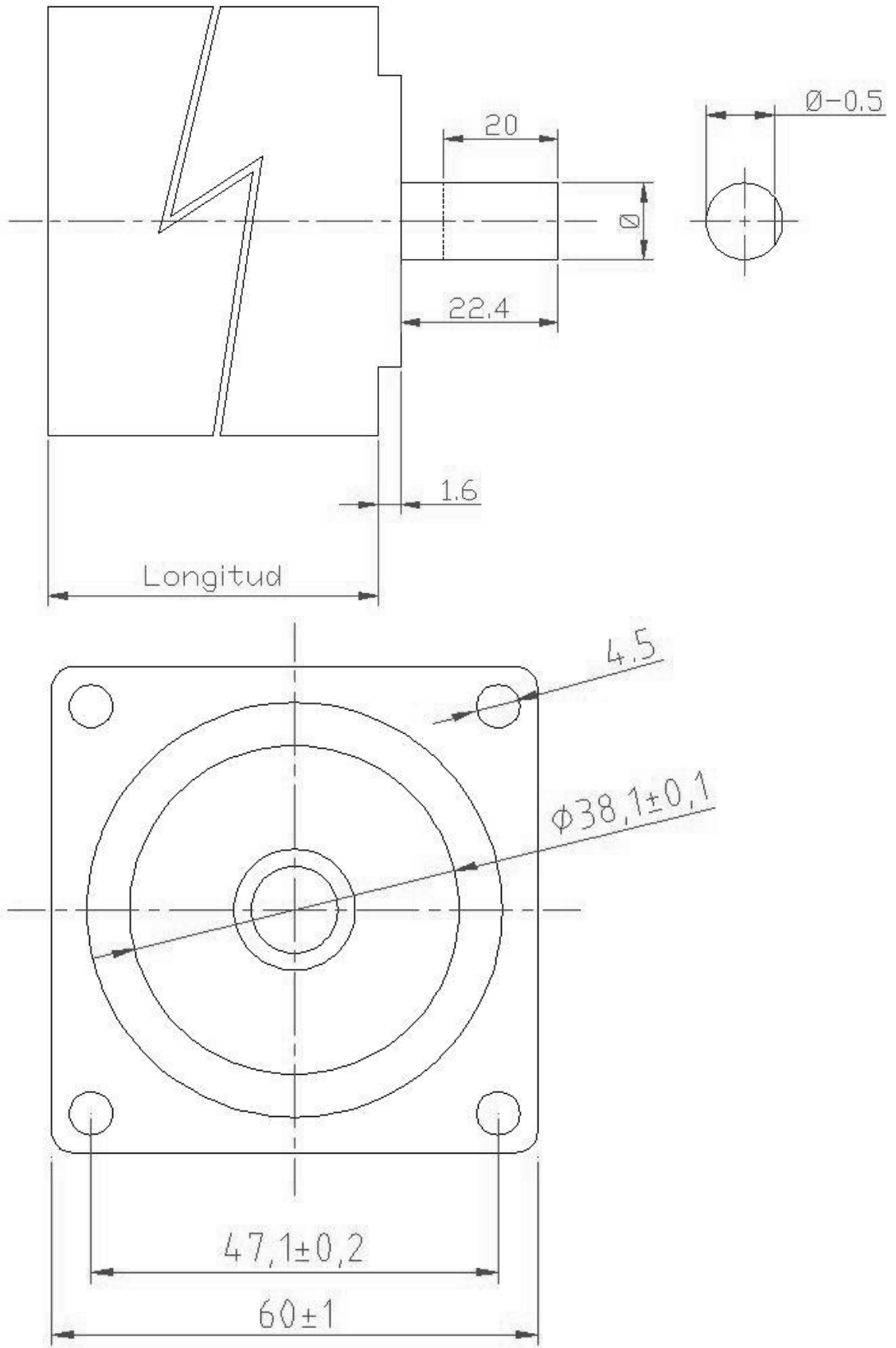
Conexiones

Bipolar Paralelo :		Motor	Controlador
		Rojo + Azul	A +
		Amarillo + Negro	A -
		Marrón + Blanco	B +
		Verde + Naranja	B -
Par máximo = Par máximo			
Corriente máximo = $2 * \text{Corriente} / \text{bobina}$			
Inducción propia = Inducción / bobina			

Bipolar Serie :		Motor	Controlador
		Rojo	A +
		Amarillo + Azul	
		Negro	A -
		Blanco	B +
		Marrón + Naranja	
		Verde	B -
Par máximo = Par máximo			
Corriente máximo = Corriente / bobina			
Inducción propia = $4 * \text{Inducción} / \text{bobina}$			

Unipolar :		Motor	Controlador
		Rojo	A +
		Amarillo + Azul	V +
		Negro	A -
		Blanco	B +
		Marrón + Naranja	V +
		Verde	B -
Par máximo = Par máximo / 1.4			
Corriente máximo = $1.4 * \text{Corriente} / \text{bobina}$			
Inducción propia = Inducción / bobina			

Dimensiones

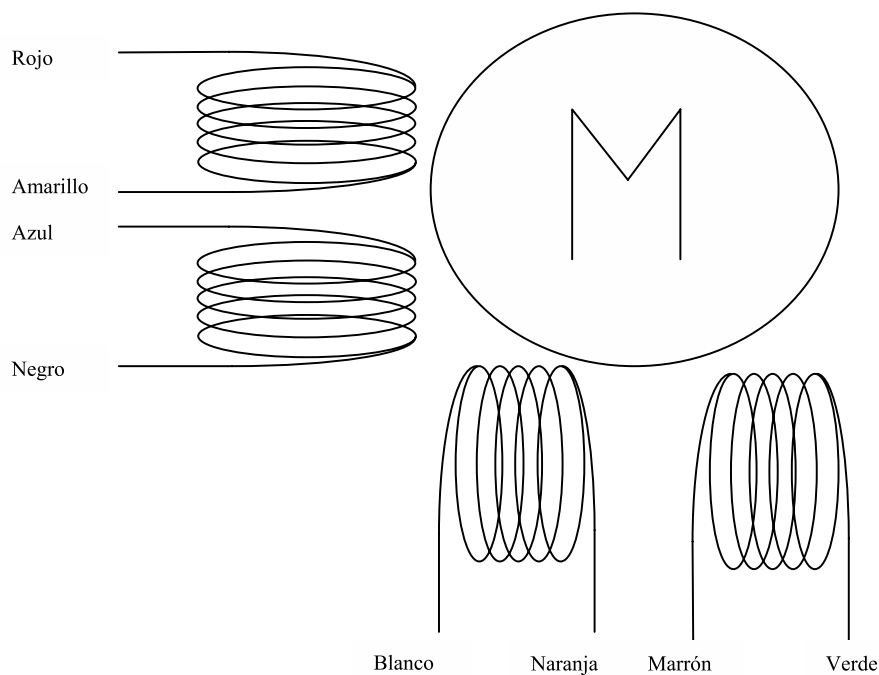


Tipo	Motor paso a paso híbrido
Normalización	Size 34 series-v2
Paso angular	1,8°
Numero de fases	2
Numero de hilos	8
ΔTmax	80 grados
Clase de aislamiento	B
Resistencia de aislamiento	100 Mohm (500 Vdc)

Tipo	Conexión	Nm	A	Ohm	V	mH	∅	L(mm)	Le(mm)	Peso (Kg)
M34-4.2-1SB12-E2-A	Paralelo	4.2	5.0	0.4	2.0	3.2	12	78	37	2.3
	Serie	4.2	2.5	1.6	4.0	12.8	12	78	37	2.3
	Unipolar	3.0	3.5	0.8	2.8	3.2	12	78	37	2.3
M34-8.5-1SB12-E3-A	Paralelo	8.5	6.0	0.5	3.0	6.0	12	115	32	3.8
	Serie	8.5	3.0	2.0	6.0	24.0	12	115	32	3.8
	Unipolar	6.0	4.2	1.0	4.2	6.0	12	115	32	3.8
M34-12.5-1SB12-E4-A	Paralelo	12.5	7.8	0.5	3.9	5.1	12	155	32	5.3
	Serie	12.5	3.9	2.0	7.8	25.4	12	155	32	5.3
	Unipolar	8.5	5.5	1.0	5.5	5.1	12	155	32	5.3

A = Corriente, Ohm = Resistencia, V = Tensión mH= Inducción propia L=Longitud

Hilos



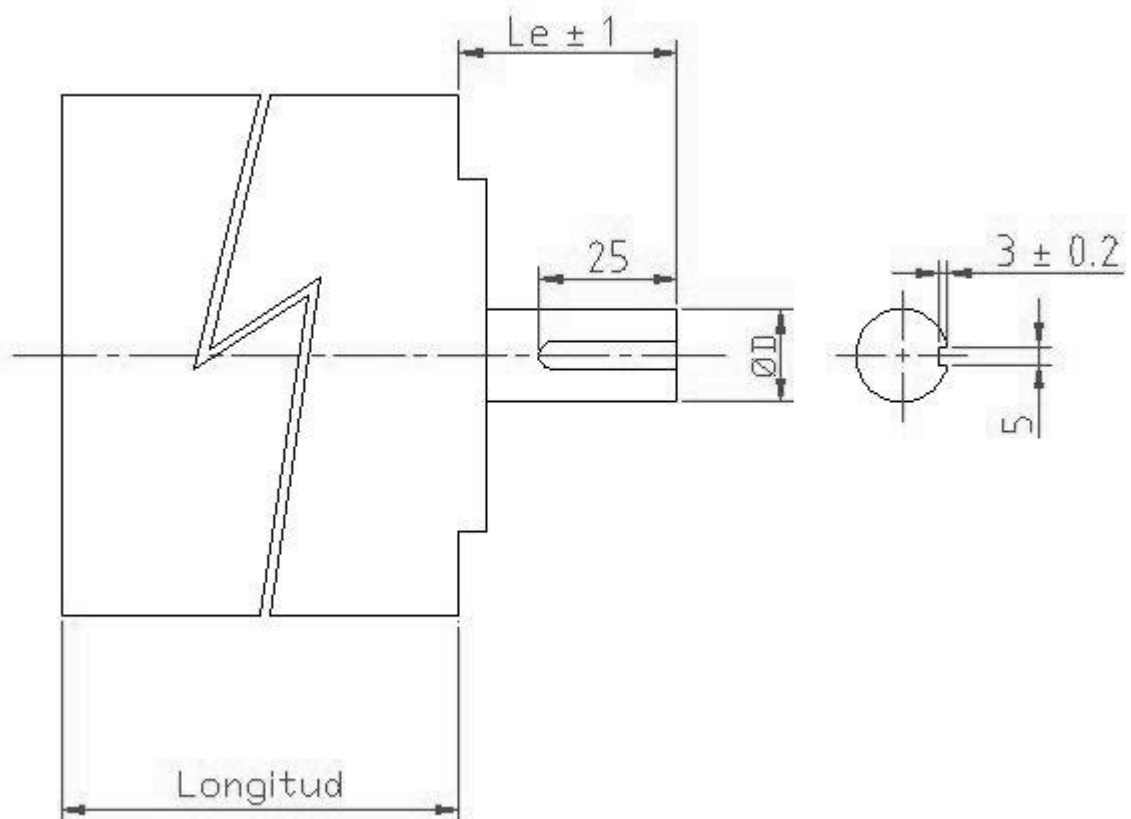
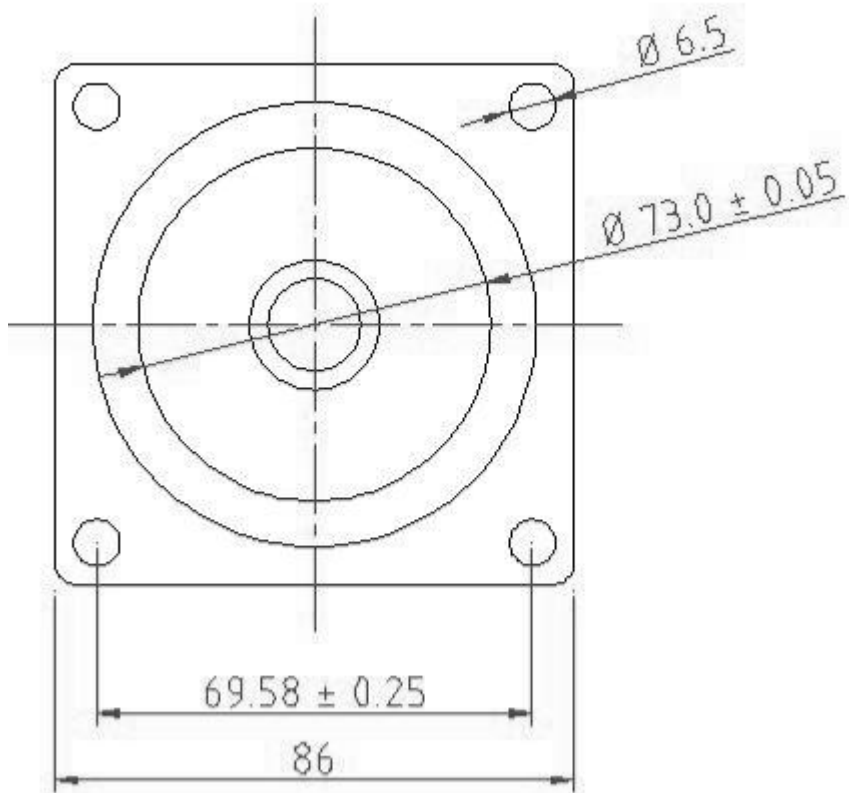
Conexiones

Bipolar Paralelo :		Motor	Controlador
		Rojo + Azul	A +
		Amarillo + Negro	A -
		Marrón + Blanco	B +
		Verde + Naranja	B -
Par máximo = Par máximo			
Corriente máximo = $2 * \text{Corriente} / \text{bobina}$			
Inducción propia = Inducción / bobina			

Bipolar Serie :		Motor	Controlador
		Rojo	A +
		Amarillo + Azul	
		Negro	A -
		Blanco	B +
		Marrón + Naranja	
		Verde	B -
Par máximo = Par máximo			
Corriente máximo = Corriente / bobina			
Inducción propia = $4 * \text{Inducción} / \text{bobina}$			

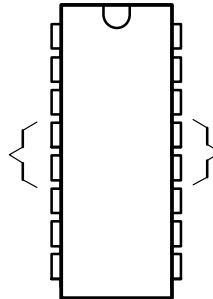
Unipolar :		Motor	Controlador
		Rojo	A +
		Amarillo + Azul	V +
		Negro	A -
		Blanco	B +
		Marrón + Naranja	V +
		Verde	B -
Par máximo = Par máximo / 1.4			
Corriente máximo = $1.4 * \text{Corriente} / \text{bobina}$			
Inducción propia = Inducción / bobina			

Dimensiones Nema34



5 Pin Configuration and Functions

NE Package
16-Pin PDIP
Top View



Pin Functions

PIN		TYPE	DESCRIPTION
NAME	NO.		
1,2EN	1	I	Enable driver channels 1 and 2 (active high input)
<1:4>A	2, 7, 10, 15	I	Driver inputs, noninverting
<1:4>Y	3, 6, 11, 14	O	Driver outputs
3,4EN	9	I	Enable driver channels 3 and 4 (active high input)
GROUND	4, 5, 12, 13	—	Device ground and heat sink pin. Connect to printed-circuit-board ground plane with multiple solid vias
V _{CC1}	16	—	5-V supply for internal logic translation
V _{CC2}	8	—	Power VCC for drivers 4.5 V to 36 V

L293, L293D

SLRS008D – SEPTEMBER 1986 – REVISED JANUARY 2016

www.ti.com

6 Specifications

6.1 Absolute Maximum Ratings

 over operating free-air temperature range (unless otherwise noted)⁽¹⁾

	MIN	MAX	UNIT
Supply voltage, V_{CC1} ⁽²⁾		36	V
Output supply voltage, V_{CC2}		36	V
Input voltage, V_I		7	V
Output voltage, V_O	-3	$V_{CC2} + 3$	V
Peak output current, I_O (nonrepetitive, $t = 5$ ms): L293	-2	2	A
Peak output current, I_O (nonrepetitive, $t = 100$ μ s): L293D	-1.2	1.2	A
Continuous output current, I_O : L293	-1	1	A
Continuous output current, I_O : L293D	-600	600	mA
Maximum junction temperature, T_J		150	°C
Storage temperature, T_{stg}	-65	150	°C

(1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, which do not imply functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions*. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

(2) All voltage values are with respect to the network ground terminal.

6.2 ESD Ratings

			VALUE	UNIT
$V_{(ESD)}$	Electrostatic discharge	Human-body model (HBM), per ANSI/ESDA/JEDEC JS-001 ⁽¹⁾	± 2000	V
		Charged-device model (CDM), per JEDEC specification JESD22-C101 ⁽²⁾	± 1000	

(1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.

(2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

6.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

		MIN	NOM	MAX	UNIT
	Supply voltage	V_{CC1}	4.5	7	V
		V_{CC2}	V_{CC1}	36	
V_{IH}	High-level input voltage	$V_{CC1} - 7$ V	2.3	V_{CC1}	V
		$V_{CC1} + 7$ V	2.3	7	V
V_{IL}	Low-level output voltage	-0.3 ⁽¹⁾		1.5	V
T_A	Operating free-air temperature	0		70	°C

(1) The algebraic convention, in which the least positive (most negative) designated minimum, is used in this data sheet for logic voltage levels.

6.4 Thermal Information

THERMAL METRIC ⁽¹⁾		L293, L293D	UNIT
		NE (PDIP)	
		16 PINS	
R_{JA}	Junction-to-ambient thermal resistance ⁽²⁾	36.4	°C/W
$R_{JC(top)}$	Junction-to-case (top) thermal resistance	22.5	°C/W
R_{JB}	Junction-to-board thermal resistance	16.5	°C/W
θ_{JT}	Junction-to-top characterization parameter	7.1	°C/W
θ_{JB}	Junction-to-board characterization parameter	16.3	°C/W

(1) For more information about traditional and new thermal metrics, see the *Semiconductor and IC Package Thermal Metrics* application report, SPRA953.

(2) The package thermal impedance is calculated in accordance with JESD 51-7.

6.5 Electrical Characteristics

over operating free-air temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT	
V _{OH}	High-level output voltage	L293: I _{OH} = 1 A		V _{CC2} - 1.8	V _{CC2} - 1.4		V	
		L293D: I _{OH} = 0.6 A						
V _{OL}	Low-level output voltage	L293: I _{OL} = 1 A			1.2	1.8	V	
		L293D: I _{OL} = 0.6 A						
V _{OKH}	High-level output clamp voltage	L293D: I _{OK} = -0.6 A		V _{CC2} + 1.3			V	
V _{OKL}	Low-level output clamp voltage	L293D: I _{OK} = 0.6 A		1.3			V	
I _{IH}	High-level input current	A	V _I = 7 V			0.2	100	μA
		EN		0.2	10			
I _{IL}	Low-level input current	A	V _I = 0			-3	-10	μA
		EN		-2	-100			
I _{CC1}	Logic supply current	I _O = 0		All outputs at high level	13	22	mA	
				All outputs at low level	35	60		
				All outputs at high impedance	8	24		
I _{CC2}	Output supply current	I _O = 0		All outputs at high level	14	24	mA	
				All outputs at low level	2	6		
				All outputs at high impedance	2	4		

6.6 Switching Characteristics

over operating free-air temperature range (unless otherwise noted) V_{CC1} = 5 V, V_{CC2} = 24 V, T_A = 25°C

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
t _{PLH}	Propagation delay time, low-to-high-level output from A input	L293NE, L293DNE				800	ns
		L293DWP, L293N L293DN				750	
t _{PHL}	Propagation delay time, high-to-low-level output from A input	L293NE, L293DNE				400	ns
		L293DWP, L293N L293DN				200	
t _{TLH}	Transition time, low-to-high-level output	L293NE, L293DNE				300	ns
		L293DWP, L293N L293DN				100	
t _{THL}	Transition time, high-to-low-level output	L293NE, L293DNE				300	ns
		L293DWP, L293N L293DN				350	

6.7 Typical Characteristics

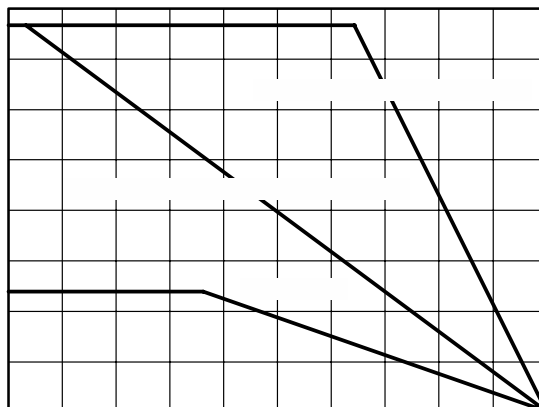


Figure 1. Maximum Power Dissipation vs Ambient Temperature