

**Posicionamiento
de un sistema multirobot
para mantenimiento
de líneas eléctricas en tensión**

Autor

Christian Müller

e-mail: christian.mueller@ifib.uni-karlsruhe.de

Madrid, Noviembre 26 de 1993

1. Introducción	1
2. Tema del trabajo	3
3. Estructura del sistema	5
3.1. Tareas del sistema / Exigencias	5
3.2. Clasificación de las operaciones	5
3.3. Componentes del sistema	5
3.4. Sistema con dos brazos	6
3.4.1. Razones para el uso de un sistema con dos brazos	6
3.5. Sistema telerobotizado	7
3.5.1. Razones para el uso de un telerobot	8
3.6. Sistema usado	10
3.6.1. Robot GRIPS de Kraft Telerobotics	10
4. Disposición de los dos robots	13
4.1. Procedimiento	13
4.1.1. Copia de un sistema ya existente	13
4.1.2. Experimentos en laboratorio o simulación	13
4.1.3. Solución analítica	13
4.2. Nomenclatura	14
4.3. Suposiciones para la simplificación	14
4.3.1. Espacio de trabajo	14
4.3.2. Posicionamiento básico respectivo al sistema de referencia	16
4.3.3. Reducción de la complejidad de los cálculos	19
4.4. Definición de medidas de evaluación	20
4.4.1. Radio de trabajo máximo para acciones coordinadas con dos brazos	21

4.4.2.	Extensión máxima para agarrar por radio de trabajo máximo r_{coord}	21
4.4.3.	Espacio de trabajo común F_{coord}	22
4.4.4.	Espacio de trabajo total F_{tot}	24
4.5.	Clasificación de las medidas encontradas	24
4.6.	Distancia mínima y máxima del sistema	25
4.7.	Distancia y ángulo óptimo considerando todas las medidas	26
4.7.1.	Unificación de las medidas de valoración.....	26
4.7.2.	Visualización gráfica	27
4.7.3.	Cálculo de la distancia óptima y del ángulo óptimo	27
4.8.	Comparación con la proporción espalda/brazo del hombre	32
4.9.	Conclusiones y posibilidad de una ampliación	33
5.	Algoritmos para la colocación asistida por computador de un sistema multirobot frente a un poste	34
5.1.	Suposiciones y simplificaciones	36
5.1.1.	Orientación del poste	36
5.1.2.	Orientación de los robots respecto al poste.....	36
5.1.3.	Ningún cambio de la colocación durante las manipulaciones	37
5.2.	Nomenclatura	38
5.2.1.	Conceptos.....	38
5.2.2.	Símbolos.....	39
5.3.	Datos necesarios	40
5.3.1.	Tipo del poste	40
5.3.2.	Posición y orientación del objeto	42
5.3.3.	Parámetros adicionales del objeto.....	44
5.3.4.	Datos geométricos del sistema multirobot.....	46
5.3.5.	Datos del espacio de trabajo	46
5.3.6.	Determinación del intervalo de la búsqueda espacial.....	48

5.3.7.	Definición de la parte del espacio de trabajo del robot que se tiene en cuenta.....	48
5.4.	Descripción del espacio de trabajo considerando orientaciones	50
5.4.1.	Ecuaciones cinemáticas del robot	50
5.4.2.	Descripción del espacio como una matriz 3D.....	54
5.4.3.	Técnicas de almacenamiento	57
5.5.	Algoritmos de colocación	62
5.5.1.	Estructura	62
5.5.2.	Preparación de los datos	63
5.5.3.	Primera prueba de plausibilidad	65
5.5.4.	Colocación inicial	74
5.5.5.	Recorrer el intervalo de búsqueda.....	75
5.5.6.	Evaluación de la colocación.....	79
5.5.7.	Colocación final	82
5.6.	Posibilidades de ampliación	83
A.	Apéndice A	85
A.1.	Representación gráfica bidimensional de las medidas de evaluación dependientes de d_R y a	85
A.1.1	Extensión máxima del agarre conjunto $d_{byp\text{eff}}(d_R, a)$	85
A.1.2	Radio máximo de trabajo $r_{coordnorm}(d_R, a)$	85
A.1.3	Proyección en xy del espacio de trabajo común $F_{coordnorm}(d_R, a)$	86
A.1.4	Proyección en xy del espacio de trabajo total $F_{totnorm}(d_R, a)$	86
A.2.	Código de MAPLE para el cálculo de parámetros d_R y a	87
A.3.	Ilustración gráfica de la determinación de d_R y a	89
A.3.1	Constante de peso $k_3 = 1$	89
A.3.2	Constante de peso $k_3 = 8$	90

B. Apéndice B	91
B.1. Método según R.P.Paul para solucionar el problema cinemático inverso	91
B.2. Código en C de las ecuaciones cinemáticas inversas para el simulador TOROS	98
B.3. Definición de los elementos mecánicos del robot GRIPS en TOROS	102
B.4. Descripción de las proyecciones del espacio de trabajo por círculos y líneas	103
B.5. Determinación de posiciones iniciales x_a e y_a	104
B.6. Ejemplo de evaluación de una colocación	105
B.6.1 Definición de los objetos y de las áreas preferidas del sistema.....	105
B.6.2 Programa MAPLE para el cálculo de la evaluación del ejemplo	106
B.6.3 Alternativa 1.....	109
B.6.4 Alternativa 2.....	110
B.7. Código de MAPLE para el cálculo del espacio de trabajo discreto	111
B.8. Visualización de espacios de trabajo para varias orientaciones	116
B.9. Representación del robot GRIPS en TOROS	118
B.10. Representación del sistema completo en TOROS	119
5. Bibliografía	120

1. Introducción

En los países modernos industrializados es un hecho que las necesidades de energía eléctrica crecerán continuamente con respecto a las necesidades de energía total. El suministro suficiente de fluido eléctrico para fabricas, centros informáticos, instalaciones de comunicación, tráfico público y además de las residencias privadas, está garantizado por un sistema sofisticado de producción y distribución de energía eléctrica. Debido a que no se pueden interrumpir procesos complejos de producción y que las malfunciones de los ordenadores y una comunicación discontinua resultarían muy costosas, hay que dar gran importancia a una alimentación *continua*. Eso significa que se realicen trabajos a las instalaciones eléctricas, si es posible, en tensión. Para esto ya existen métodos y herramientas como, por ejemplo, utilizar puentes de bypass, trabajar con pértigas o plataformas aisladas. Todos estos trabajos, aún son hechos manualmente, son peligrosos y gastan mucho tiempo por el empleo de gran seguridad. Bajo este contexto IBERDROLA y DISAM desarrollan un proyecto de investigación denominado ROBTET, cuyo objeto es el desarrollo de un sistema robotizado competitivo, que es capaz de realizar trabajos de mantenimiento a instalaciones eléctricas hasta 20 kV. Sistema similares ya son desarrollados en Japón y EE UU.

Estas investigaciones han demostrado que un sistema robotizado reduce drásticamente la duración de trabajos para tareas especiales.

Para aprovechar el tiempo del desarrollo tan efectivamente como sea posible y para disponer pronto de un prototipo del sistema funcionado, se realizó una clasificación del proyecto ROBTET, que complementa el sistema después de cada nivel del desarrollo con un componente adicional. Esta estructuración es explicada brevemente, para poder clasificar el trabajo presente en el contexto general.

En el primer paso del desarrollo se hace la determinación, selección e integración de cada uno de los componentes de un prototipo (*Integración de sistemas*). Eso incluye el hardware completo (grúa, robot, sistemas de alimentación, ...) y su disposición básica para su uso.

A diferencia de los sistemas americano y japonés, que ofrecen como sistemas adicionales de apoyo para el usuario solamente cámaras y monitores, se intenta crear aquí un sistema multimedia como *interfase hombre-maquina*. Se usa la

integración de sensores y cámaras para el modelado del entorno en una forma adecuada. Con estos datos y una presentación gráfica inteligente se puede ayudar intensivamente al usuario en la realización de sus trabajos (*Telepresencia*).

Se puede denominar la evolución y ampliación de la telepresencia orientada a un apoyo activo e inteligente como *teleoperación protegida*. Aquí se verifica, si el usuario realiza operaciones en zonas peligrosas (dentro de la distancia de seguridad) o si ocurren colisiones con objetos del entorno. Para evitar esas colisiones o para recomendar un acceso apropiado se aprovechan algoritmos especiales de planificación de trayectorias y verificación de colisiones.

Las operaciones que aparecen con frecuencia en un orden fijo (operaciones rutinarias), deben ser realizadas por el sistema independientemente del usuario (*Automatización de las operaciones*). Mientras se ejecuta se puede, por ejemplo, preparar la tarea próxima. Ya que se trata de un sistema de múltiples robots, se deben implementar métodos de manipulación coordinada.

Hasta ahora se han utilizado sólo componentes estándar, que por cierto se pueden usar muy flexiblemente, pero con frecuencia con un empleo mayor de programación y planificación (división secuencial de acciones complejas). Con un *desarrollo de herramientas especiales*, incluyendo un sistema de cambio automático, se puede facilitar estas manipulaciones enormemente. Además de esto se desarrollan *programas adicionales* para resolver problemas específicos realizando la tarea mediante robots.

El último período del desarrollo prevé la integración de hardware adicional, con los interfases apropiados al sistema básico, lo cual es añadido para completar el sistema en casos especiales (*sistemas auxiliares*).

2. Tema del trabajo

El Proyecto Fin de Carrera se divide en dos partes que se enmarcan dentro de los apartados de *Integración de sistemas* y *Desarrollo de herramientas y aplicaciones*. El elemento que une las dos secciones es la consideración y la descripción del espacio de trabajo del robot y el siguiente aprovechamiento de los conocimientos adquiridos.

Al contrario de la situación con robots individuales, la disposición relativa de los robots en un sistema multirobot no es un problema trivial. En caso de robots que trabajan separadamente se puede considerar cada robot aparte en su entorno y sólo se tiene que garantizar que no hay colisiones. Pero en este caso los robots operan sobre un espacio de trabajo común, lo que posibilita acciones coordinadas. El objeto de trabajo es entonces, considerando condiciones específicas y la estructura del sistema, encontrar la mejor disposición de los robots. Se prefiere conscientemente formular el problema analíticamente en comparación con la búsqueda experimental, para conseguir una solución general que se puede modificar luego fácilmente.

Dada la diversidad de escenarios, motivado por el gran número de postes y trabajos realizados, no es fácil determinar una colocación satisfactoria del sistema relativa al poste. Debido a la distribución tridimensional de los objetos, el posicionamiento manual resulta muy difícil. Una realización de los trabajos sin una pérdida de tiempo grande, sólo es posible con pocos cambios de la colocación, considerando también que la determinación del sistema de referencia cuesta mucho tiempo. Por eso, el enfoque de las consideraciones es desarrollar una estructura y los algoritmos para un programa, que facilite la elección rápida de una colocación, partiendo de los objetos y parámetros especiales como, por ejemplo, la geometría del poste o la distancia de seguridad. Este programa puede ser utilizado directamente in situ, o como herramienta de planificación integrada en un simulador para robots, apriori en el laboratorio. Por eso el trabajo incluye también el modelado del sistema en TOROS [9].

Para ninguno de estos dos problemas se podría recurrir a soluciones estándar ya existentes. La descripción y formulación de espacios de trabajo de robots todavía no han sido resueltas en general para cualquier aplicación. El estudio de la bibliografía ha resultado muy difícil, porque cada uno de los autores hace bastantes

restricciones para su tarea especial. Por eso, se han desarrollado métodos originales para solucionar los dos problemas planteados. Debido al tiempo limitado para este diplomatura, no se implementó el programa entero. Sin embargo se han verificado y visualizado partes importantes de los algoritmos con el programa MAPLE [10] suponiendo ejemplos simples.

3. Estructura del sistema

3.1. Tareas del sistema / Exigencias

Este sistema debe ser utilizado especialmente para mantenimiento, inspección y reparación de líneas eléctricas en tensión. Está previsto inicialmente su empleo en España, cuya estructura de la red de distribución de energía eléctrica ofrece ciertas ventajas para la aplicación económica y sencilla de este sistema robotizado. La poca altura de los postes (12m-15m) y una tensión relativamente baja (20kV) son restricciones que reducen la complejidad del sistema respecto al aislamiento y a las medidas de seguridad.

Sin embargo, no es posible determinar la especificación en algunos criterios, en razón al gran número de postes y de tareas distintas. Tampoco se pueden pronosticar todos los tipos de daños, ni se tiene la capacidad de proyectar todos los detalles de las tareas. Por eso hay que intentar, realizar la integración del sistema muy flexiblemente y de aplicación universal, para proporcionar suficientes grados de libertad y facilitar la adaptación del sistema.

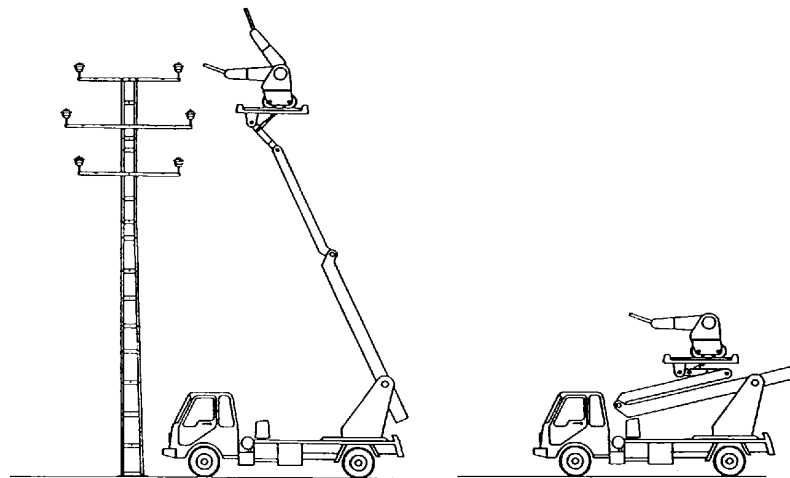
3.2. Clasificación de las operaciones

En la primera fase del proyecto se investigaron los métodos de los trabajos en tensión y se realizó una estructura jerárquica. Se hizo una sistematización que se refería a los elementos manejados (aislamiento, derivaciones, etc.). Esta nuevamente se divide en acciones específicas (conectar, ...) sobre los objetos respectivos. Ya que estas acciones son demasiado complicadas para un sistema robotizado se redujo finalmente la división en primitivas, es decir comandos muy sencillos que pueden ser entendidos directamente por el sistema (tomar, mover, girar, etc).

3.3. Componentes del sistema

Para la primera definición de los componentes se había utilizado la clasificación anterior y se determinaba la configuración en razón de las acciones a realizar. Esta

prevé una grúa con dos robots que son montados encima de una plataforma. Los robots tienen una gran exactitud pero relativamente poca capacidad de carga. Para objetos pesados hay adicionalmente un brazo con una alcance y capacidad de carga grande. Todo el sistema es equipado con sensores, cámaras, un sistema multimedia y con un sistema de control que permite tanto teleoperaciones como operaciones automáticas y tareas de inspección. La figura siguiente muestra un bosquejo del sistema.



Diseño del sistema

3.4. Sistema con dos brazos

3.4.1. Razones para el uso de un sistema con dos brazos

Sistemas con un brazo robotizado se han establecido desde hace mucho tiempo en la industria. Se los utiliza principalmente en la fabricación para trabajos de montaje, soldadura y ensamblado. Casi todos los modelos del mercado se encuentran con especificaciones de velocidad, exactitud y programación en un nivel muy alto. Desgraciadamente estos sistemas con un brazo no se pueden aplicar siempre a tareas solucionables con dos brazos. Por eso en el área de la automatización industrial y también en la técnica nuclear o en la investigación espacial se desarrollan y usan más frecuentemente sistemas con brazos coordinados.

3.4.1.1. *Problema del cambio de la posición de agarre*

Un sistema con un brazo no puede recoger un objeto sin apoyo de una instalación adicional, que puede ser desde un utillaje sencillo hasta un sistema de manipulación complejo. Esto exige un entorno del sistema muy especializado y adaptado, lo que en nuestro caso no puede ser instalado por el gran número de objetos distintos. Una forma más elegante y más sencilla es el cambio de la posición con ayuda de dos brazos, donde un brazo posiciona el objeto en una manera apropiada que resulta óptima para ser cogido por el otro brazo.

3.4.1.2. *Movimientos más complejos*

Al manipular objetos pesados resultan fuerzas grandes en las articulaciones y la muñeca del robot, razón por la cual sólo pueden ser realizadas operaciones restringidas. En comparación, el sistema de dos brazos puede, por ejemplo, apoyar en uno la manipulación del otro. Además tareas como juntar, torcer y sacar no serán un problema para esa configuración.

3.4.1.3. *Mayor espacio de trabajo*

Aunque este punto parece muy claro, hay que pensar que la utilización de más robots individuales en un mismo espacio de trabajo podría eventualmente reducirlo. Sin embargo se puede ampliar el espacio de trabajo con una disposición inteligente de los robots y también el alcance de sistema crecerá enormemente en algunas direcciones.

3.4.1.4. *Similitud a la forma de trabajo humano*

El desarrollo de manipuladores sencillos hasta robots complejos se orienta muchas veces al cuerpo humano que ofrece la mejor movilidad y flexibilidad con respecto a manipulación de objetos. Un punto muy importante es además el manejo de estos manipuladores, que puede ser aprendido más rápida, fácil y exactamente cuanto más se oriente el sistema al hombre.

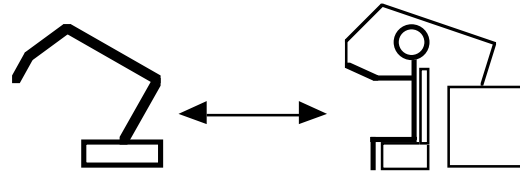
3.5. Sistema telerobotizado

Los sistemas telerobotizados apoyan la realización de tareas especiales en entornos difíciles e inaccesibles, así como allí donde resultan riesgos directos para el usuario.

Hay que mencionar especialmente aquí aplicaciones de instalación en el espacio, en la técnica nuclear y para tareas submarinas. Tiene sentido clasificar los sistemas telerobotizados para aclarar los pasos del desarrollo y los componentes de sistemas distintos[v.fig.].

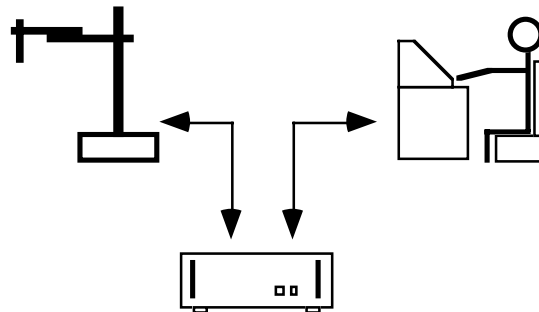
Telepresencia

El hombre controla **completamente** el manipulador (*esclavo*) mediante una imitación del mismo pero adaptado en tamaño (*maestro*). Por la realimentación de los señales **visuales y táctiles**, el usuario tiene el sentimiento de realizar la manipulación directamente. Se realiza la programación con el uso de sistemas reflejados '*maestro-esclavo*'.



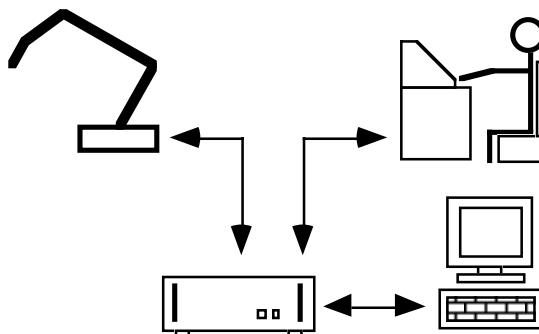
Teleoperación

En este caso el hombre también controla **completamente** las acciones. Los manipuladores no se asemejan al hombre y tampoco la instalación de control. Las **transformaciones de coordenadas** que se necesitan son realizadas por un ordenador. Además el sistema posee la posibilidad para adquirir y almacenar datos de la acción para repetirla otra vez.



Telerobot

Aquí el hombre controla al robot en **modo teleoperado**. En el **modo automática** pueden ser realizadas operaciones simples. Tras acabar estas manipulaciones, el sistema vuelve automáticamente al modo teleoperado. Por el carácter híbrido se le denomina sistema telerobot.



La clasificación exacta de sistemas del mercado dentro de uno de estos tres sistemas no es posible hoy. Casi todos se componen de una mezcla de estos tipos básicos.

3.5.1. Razones para el uso de un telerobot

A continuación se explican algunas razones de porque no se implementa desde el principio un robot convencional sino un sistema telerobotizado. En cualquier caso se reserva la opción de la automatización para el desarrollo posterior.

3.5.1.1. Entornos y tareas cambian frecuentemente

Ya que hay tareas y entornos de trabajo distintos es solo posible proyectar todas las tareas, es decir las manipulaciones y movimientos, con un gasto de tiempo y programación enorme.

3.5.1.2. Reacción a sucesos imprevisibles

Un objetivo del proyecto es habilitar la realización de trabajos de mantenimiento los días, en que el tiempo prohibiría la realización manual. Por ejemplo el viento fuerte movería los cables y la grúa. Esto causa que el modelo del mundo almacenado y el mundo real no coincidan e impide una operación eficaz del robot.

3.5.1.3. Determinación difícil del sistema de referencia

Los dos robots son montados encima de una plataforma, que es posicionada por una grúa. En razón de la precisión baja de la grúa y de la posibilidad de vibración, conseguir una colocación exacta es muy difícil. Aquí es necesario adaptar el modelo del entorno continuamente al entorno real, lo que exige un sistema de sensores sofisticado y además una nueva simulación de las tareas. Esta simulación destruye posiblemente la ventaja de tiempo reduciendo la rentabilidad. Sería imaginable aquí un anclamiento de la plataforma al poste, lo cual aumentaría el empleo del aislamiento y dificultaría la mecánica.

3.5.1.4. Un sistema completamente automatizado es costoso y poco robusto

Para que los robots realicen sus tareas completamente automatizadas, se tendría que desarrollar un sistema de sensores extraordinariamente sofisticado. Los sistemas de proceso de imagen tienen todavía restricciones grandes porque por ejemplo, suciedad y humedad dificultan mucho el reconocimiento correcto de los objetos. Prescindiendo de sistemas de medidas laser es muy difícil alcanzar la robustez y la compatibilidad electro-magnética.

Hay que proteger todo el equipo electrónico (ordenadores para el proceso de imagen, sistema de control, sensores,...) de las influencias del tiempo. Todo eso aumenta la complejidad del sistema y cuando falla un componente del sistema, el resto queda paralizado.

3.5.1.5. *Sistema fácil de aprender y utilizar*

Debido a que la parte de los costos de personal crece permanentemente respecto a los costes totales, es importante para el uso económico del sistema que pueda ser utilizado no sólo de especialistas bien entrenados, sino también por los operarios que hacen hoy el trabajo manualmente. Así se puede ahorrar el tiempo de formación para manejar el sistema.

Además hay que tener en cuenta que hoy en día cada decisión sobre la automatización de un proceso, que se ha hecho hasta ahora manualmente de hombres, no solo debe considerar aspectos financieros y tecnológicos sino también sociales.

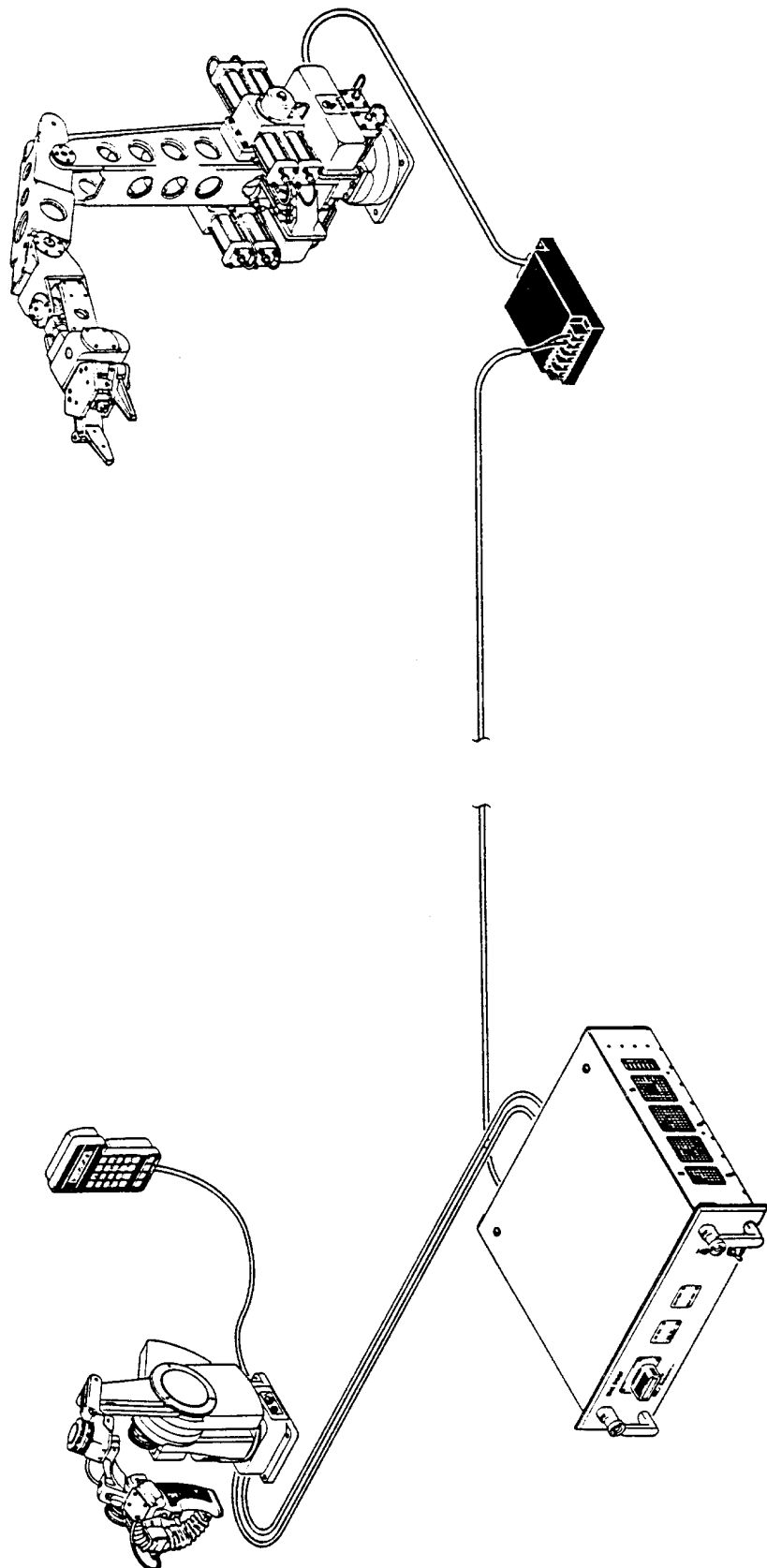
3.6. Sistema usado

3.6.1. *Robot GRIPS de Kraft Telerobotics*

El sistema telerobotizado no se puede clasificar exactamente en el orden presentado anteriormente. Se compone de un brazo electrohidráulico (*esclavo*), con capacidad de carga, exactitud y velocidad grandes. Por el accionamiento hidráulico, el sistema es relativamente ligero y robusto, por eso permite trabajar en presencia de campos electromagnéticos fuertes. Experimentos correspondientes a tensiones hasta de 300 kV fueron hechos con éxito en el proyecto americano TOMCAT que utiliza también el robot GRIPS. Especialmente apropiada parece la posibilidad de controlar directamente la velocidad de cierre y la fuerza de agarre de la pinza así como la existencia de dos modos de utilización: En el modo *slave-wrist* la pinza es controlada directamente por los movimientos del usuario y en la modo continuo se puede girar continuamente controlando la velocidad de la muñeca, permitiendo atornillar o enrollar. El brazo *master*, que supone por su realimentación de fuerza y su ergonomía una amigable interfase hombre-maquina, ofrece todas las facilidades de un uso intuitivo. El sistema de control esta montado modularmente, compuesto por una unidad CPU central y una unidad remota protegida situada en el robot. Las dos unidades pueden ser conectadas vía cable paralelo, cable coaxial o fibra óptica. El sistema de control ofrece además:

movimiento escalado	graduación de las relaciones de movimientos del <i>maestro-esclavo</i> , p.e. para un posicionamiento final exacto
movimiento indexado	Offset del master relativo al <i>esclavo</i> , para aumentar el confort del uso adaptando el <i>maestro</i> al usuario
movimiento restringido	determinación vía software de limites para ejes particulares así como el robot entero
aprendizaje	aprender secuencias de acciones, que son almacenadas y pueden ser repetidas. Se almacenan las tareas respectivamente con un número de identificación propio en una librería y se pueden realizar hacia adelante y hacia atrás a varias velocidades
detención de movimiento	Congela la posición actual de los ejes en cualquier momento
bloqueo de ejes	Bloqueo de ejes unicos

Kraft
TeleRobotics, Inc.



4. Disposición de los dos robots

4.1. Procedimiento

En principio se pueden utilizar tres métodos distintos para la determinación de la disposición, sin considerar la disposición aproximada por sentido común.

4.1.1. *Copia de un sistema ya existente*

Esta posibilidad ni es muy sofisticada ni tiene sentido porque hay muy pocos sistemas similares sobre los que se requerirían demasiadas suposiciones y restricciones y así no se admitiría la introducción de conceptos nuevos. Además a partir de una consideración propia se obtienen condiciones nuevas que infuyen en otras partes del proyecto.

4.1.2. *Experimentos en laboratorio o simulación*

Este camino, que fue seguido por el sistema japonés, prevé la creación de un laboratorio con una imitación real del entorno y del sistema robotizado (plataforma, líneas eléctricas, postes, robots, ...). Tanto este procedimiento como una simulación requieren mucho tiempo y se debe determinar primero muy exactamente la especificación de las tareas, los postes y las herramientas. Esto conduce a una solución válida para los escenarios probados, pero es muy poco general y difícilmente susceptible a modificaciones.

4.1.3. *Solución analítica*

Este procedimiento ofrece, al contrario de los anteriores, las ventajas siguientes:

- barato, porque se puede renunciar al montaje de un entorno de pruebas y no se necesita tanto tiempo
- lleva a una solución exacta y clara
- ya que no se consideran exclusivamente tareas y geometrías de postes especiales se consigue una solución más general, que deja libertad para la integración de nuevas técnicas de mantenimiento, herramientas y tipos de postes
- facilita la implementación en un ordenador situado directamente in situ, para adaptar la disposición de los robots rápidamente a situaciones nuevas.

Naturalmente (hubiera sido muy hermoso) hay también algunas desventajas, que resultan en gran parte por la descripción matemática de la cooperación de los robots. Además hay que hacer muchas suposiciones y simplificaciones para conseguir un volumen de cálculos manejable.

4.2. Nomenclatura

S_0	sistema de referencia de la base
S_L	sistema de referencia del robot de la izquierda (positivo en y)
S_R	sistema de referencia del robot de la derecha (positivo en y)
d_R	distancia entre los robots ($ S_R - S_L $)
a	ángulo de orientación de los robots (respecto al eje z)
r_{coord}	radio máximo de trabajo
d_{byp}	Extensión maxima de agarre conjunto en el radio de trabajo maximo r_{coord}
F_{coord}	proyección en xy del espacio de trabajo común
F_{tot}	proyección en xy del espacio de trabajo total
$r_{coordnorm}$	medida normalizada de r_{coord}
$d_{bypnorm}$	medida normalizada de d_{byp}
$F_{coordnorm}$	medida normalizada de F_{coord}
$F_{totnorm}$	medida normalizada de F_{tot}
d_{bypeff}	distancia efectiva de agarre conjunto ($d_{bypnorm} \times r_{coordnorm}$)
k_1	constante de peso para d_{bypeff}
k_2	constante de peso para $r_{coordnorm}$
k_3	constante de peso para $F_{totnorm}$
k_4	constante de peso para $F_{coordnorm}$

4.3. Suposiciones para la simplificación

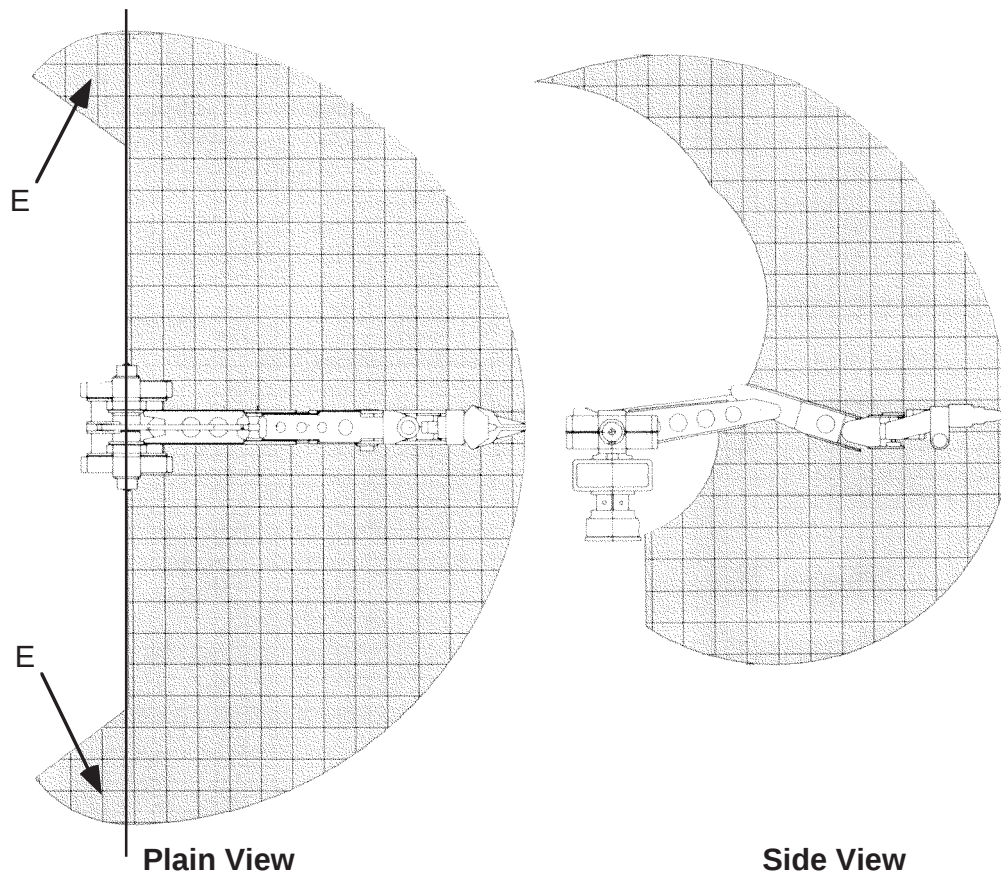
4.3.1. *Espacio de trabajo*

Las posibilidades de la utilización de un robot y su potencia para realizar manipulaciones en su entorno, dependen en gran medida del tamaño de su espacio

de trabajo. Por esto, se estudia primero el espacio de trabajo de los dos robots, antes de pensar en la posición del sistema total o su programación.

4.3.1.1. Datos conocidos

En la descripción técnica del robot GRIPS hay solo una representación del espacio de trabajo con cortes en dos planos distintos del espacio [v.fig.].

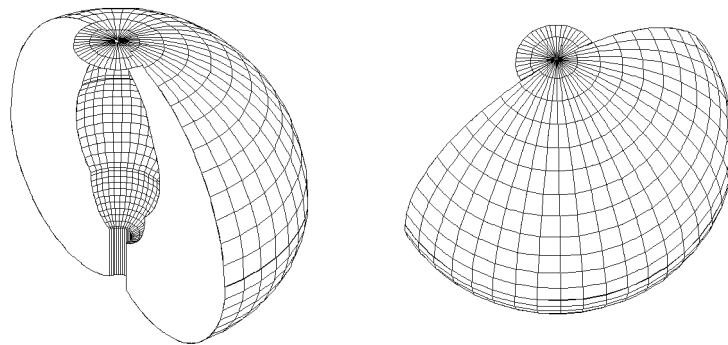


Representación del espacio de trabajo en dos vistas

Para determinar a partir de los cortes el espacio asequible hay que ampliar la vista lateral generando un cuerpo por rotación. Entonces hay que intersectarlo con el cilindro que se forma de una ampliación en la tercera dimensión de la otra vista. Esto es posible utilizando programas CAD capaces de modelar sólidos. Se puede representar así pues el espacio de trabajo gráficamente y también calcular su volumen.

4.3.1.2. *Visualización del espacio de trabajo*

La visualización siguiente del espacio de trabajo fue creada con un programa de CAD (AutoCad) y da una idea del procedimiento para generar el cuerpo por rotación [v.fig.]. Además se supone que los puntos alcanzados por el robot están siempre dentro del plano determinado por la rotación de su primer eje (robot es planar), y por eso no se tienen en cuenta las esquinas de la vista en planta E.



Visualización tridimensional del espacio de trabajo

Sin embargo es mas importante el problema de la determinación de la posición y orientación relativas de los robots, para lo cual esta representación nos da solo una idea. Por lo tanto es importante buscar una medida dependiente de d_R y a , para evaluar la situación de trabajo de ambos robots en nuestras tareas particulares.

4.3.2. ***Posicionamiento básico respectivo al sistema de referencia***

En la siguiente contemplación es necesario detallar un poco la descripción del espacio de trabajo, para lo cual se utiliza la clasificación propuesta por Becquet. Su definición general del espacio de trabajo de un robots es:

- El espacio de trabajo es el espacio de todas las posiciones y orientaciones del TCP, donde existe una solución para el problema cinemático inverso.

A partir de esto se hace la diferenciación siguiente:

4.3.2.1. *Espacio de trabajo asequible*

Aquí es interesante considerar todas las posiciones posibles, sin contemplar la orientación. Es decir que se incluyen también puntos que son asequibles solo en

una orientación y que para algunas manipulaciones practicas son inutilizables. Se renuncia por tanto a grados de libertad.

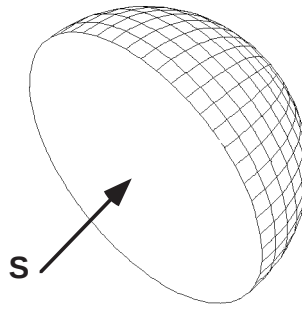
4.3.2.2. Espacio de trabajo practico

Aquí se tienen en cuenta solo aquellos puntos, que se pueden alcanzar en todas las orientaciones posibles. Este espacio es naturalmente mas pequeño pero de mucho mas valor para manipulaciones.

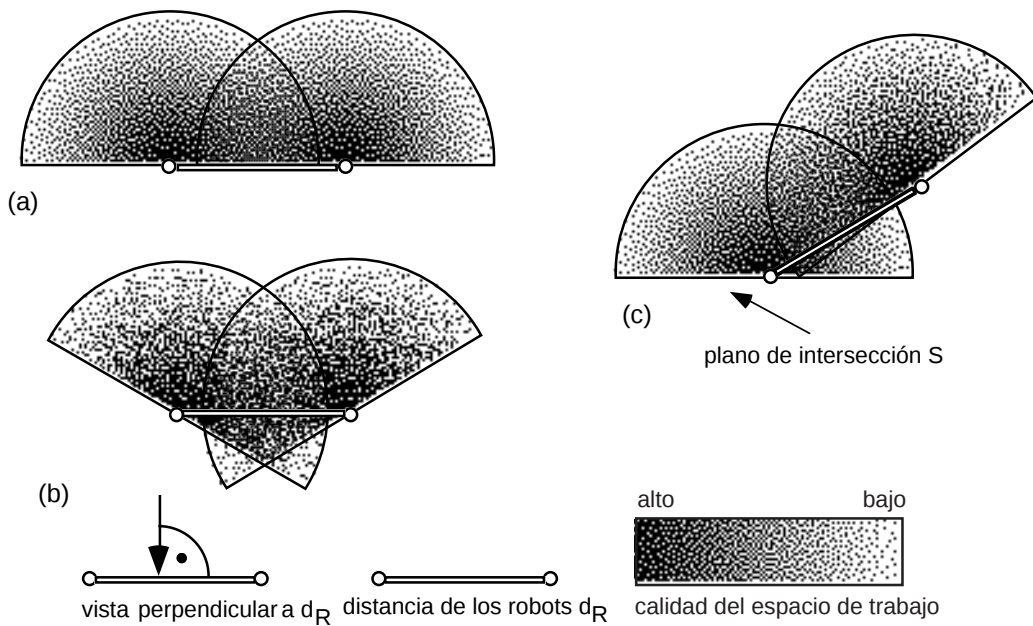
El espacio de trabajo asequible representa para una estimación gruesa un máximo del tamaño de espacio y el espacio de trabajo práctico un mínimo. La 'calidad' del espacio de trabajo, comparado con estos extremos, es inversamente proporcional. La calidad disminuirá seguramente en dirección al borde, porque los puntos pueden ser alcanzados en menos orientaciones.

4.3.2.3. Determinación de las variables que describen la disposición

Si se supone simplemente, que la calidad se disminuye radialmente en un espacio con forma de media esfera, entonces la colocación de los robots en una distancia absoluta d_R con los planos de intersección S parcialmente superpuestos produce un máximo de espacio común utilizable[v. fig. (a)]. La palabra 'utilizable' se refiere al hecho que el espacio entre los dos robots tiene solo poco valor para nosotros, porque los objetos manipuladas no son implantadas de fuera si no tienen que ser alcanzadas de los robots en una distancia tan grande como sea posible. Las otras posibilidades de la disposición, las que tienen ventajas para otras tareas (robots coordinadas en una área de montaje), aquí no tienen sentido. Así se reconoce por ejemplo según fig. (b) muy bien que una grande parte del espacio de trabajo se encuentra detrás de los robots, y que una disposición según fig. (c) haría el espacio de trabajo del robot de atrás parcialmente inutilizable, porque el robot delantero le impide el acceso. Además se acentuaría con una colocación no simétrica uno de los robots, cual no seria recomendable porque la simetría de los postes.



Aproximación del espacio como media esfera



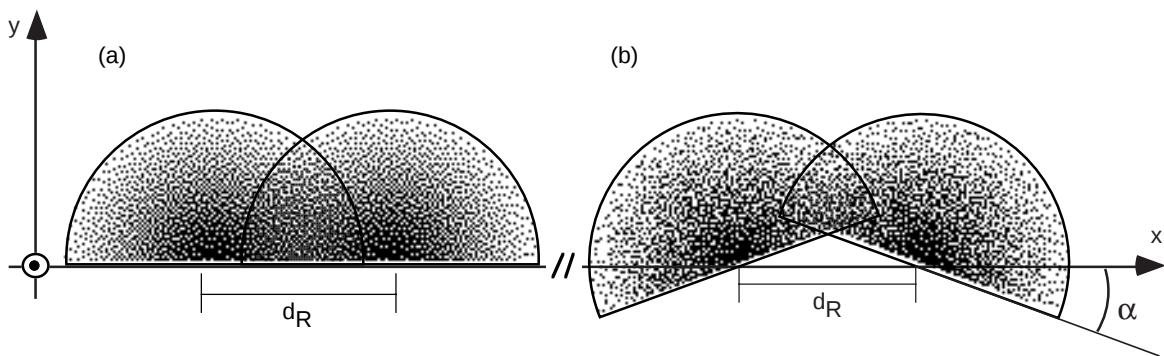
Vista en planta de varias disposiciones

Partiendo de estas reflexiones y del hecho que los robots tienen que ser montados perpendicularmente (ejes-z de S_L , S_R y S_0 paralelos) se hacen las determinaciones siguientes:

- Las configuraciones básicas de ambos robots son iguales, es decir que en sus posiciones cero sus brazos son paralelos al eje y de S_0 .
- Los robots se encuentran en el eje x a una distancia de d_R .

Esta orientación ofrece el espacio común mas grande con alta calidad. Seguramente esta orientación es además la mas intuitiva para el usuario, porque es similar a la disposición del hombre [v.fig. (a)].

Por el momento d_R es la única variable, que determina la posición relativa de los robots. Ya como dicho antes, el espacio entre las bases de los robots es poco utilizado. Por eso no se acepta la suposición del paralelismo de los brazos y se define un angulo de giro de los robots hacia afuera. En principio se pierde espacio de trabajo común con baja calidad, pero se gana espacio de trabajo total, que se puede utilizar por ejemplo para el cambio de herramienta o para recoger objetos.[v.fig.(b)].

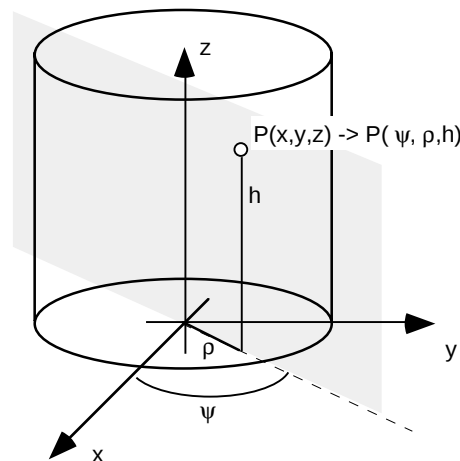


Representación de los parámetros d_R y α

4.3.3. Reducción de la complejidad de los cálculos

A causa de la dificultad para describir matemáticamente el espacio de trabajo tridimensional, solo se contemplará la proyección en el plano en xy, para convertir este problema en uno bidimensional. Las reflexiones siguientes muestran, que esto es una buena aproximación:

Si se divide el espacio de trabajo tridimensional en dos proyecciones en xy y en yz, cualquier punto alcanzable por el robot debe cumplir con un criterio suficiente y uno necesario. El punto tiene que encontrarse obligatoriamente en el plano xy de S_L/S_R y se debe verificar, via una transformación en coordenadas cilíndricas, si el punto se encuentra también en la proyección sobre el rz de S_L/S_R . 'Suficiente' significa aquí que un punto en dentro de la proyección yz puede, pero no tiene que encontrarse en el corte en rz.



Sistema de coordenadas cilíndricas

Las dos variables dR y α están solo definidas en el plano xy y no cambian la orientación del espacio de trabajo en la dirección z (el vector normal de la proyección queda paralelo al eje z). Por eso es válido, hacer contemplaciones referentes a dR y α en xy , porque el tamaño del volumen está determinado por el plano definido.

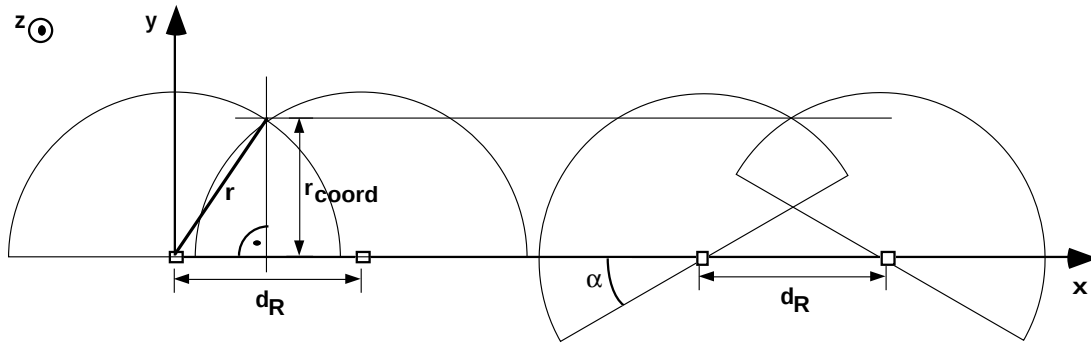
4.4. Definición de medidas de evaluación

Después de hacer algunas definiciones y simplificaciones hay que definir ahora un conjunto de medidas, que dependan de la distancia dR y el ángulo α , con las cuales sea posible hacer un análisis cuantitativo. Se podrían contemplar las acciones específicas (abrir o cerrar puentes, seccionar, conectar,...) para derivar medidas valorables. A causa de la multitud de estas acciones y del hecho que un cambio de la técnica del trabajo o la introducción de nuevas herramientas, no garantiza la validez del análisis, no se intenta considerar ninguna tarea especial sino que se obtienen expresiones valorables en casos más generales. Esto garantiza también una mayor flexibilidad del sistema.

4.4.1. Radio de trabajo máximo para acciones coordinadas con dos brazos

La potencia de un sistema con dos brazos se funda principalmente en la posibilidad de manipular con un brazo un objeto mientras el otro le apoya para, por ejemplo fijar o girar el objeto.

Suponemos aquí que el objeto tiene mas o menos la misma dimensión que la pinza y que no tiene longitudes grandes en ninguna dirección. Puesto que estas manipulaciones deben ser ejecutadas a una distancia de la plataforma posiblemente grande (distancia de seguridad, geometría desfavorable del poste), es importante maximizar la distancia a puntos que sean asequibles para ambos brazos al mismo tiempo. Partiendo de la representación del corte xy, como medios círculos, se pueden hacer contemplaciones geométricas sencillas.[v.fig.].



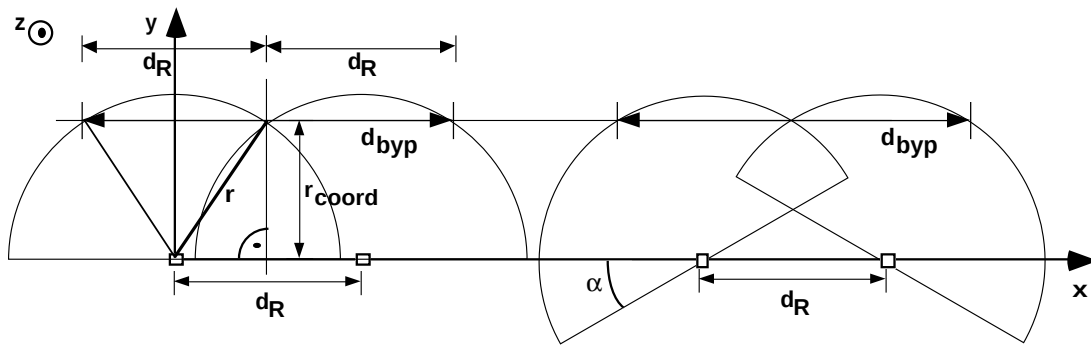
Radio de trabajo máximo r_{coord}

Se observa, que r_{coord} no depende de a . Por lo tanto la dependencia matemática de d_R esta dada por:

$$r_{coord}(d_R) = \sqrt{r^2 - (d_R/2)^2}$$

4.4.2. Extensión máxima para agarrar por radio de trabajo máximo r_{coord}

En la definición de r_{coord} fue hecha la restricción de una dimensión pequeña para los objetos manipulados (objetos compactos). Pero para tareas particulares (bypass, derivaciones, fijar aislamientos,...) hay que trabajar con objetos largos y grandes. A partir de esto se introduce una medida, cuyo significado se expone con mayor claridad gráficamente[v.fig.]. En este caso también se deben realizar los trabajos en las líneas en tensión posiblemente lejos, a causa de la distancia de seguridad. A diferencia de r_{coord} , se exige simultáneamente una extensión máxima para agarrar a esta distancia. Además de esta forma resulta un espacio grande para mover los objetos.



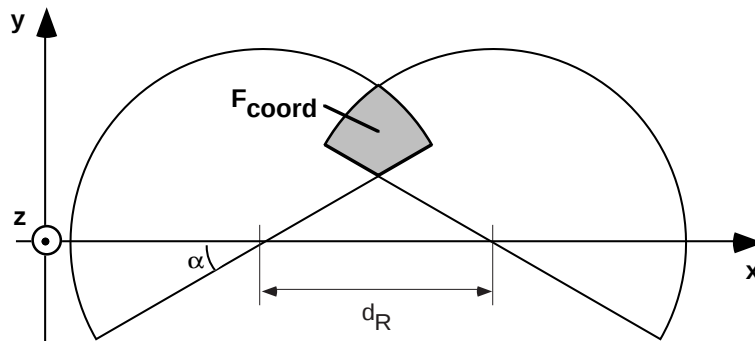
Extensión máxima de agarre d_{byp}

La contemplación geométrica da la dependencia siguiente de d_R :

$$d_{byp}(d_R) = 2 \cdot d_R$$

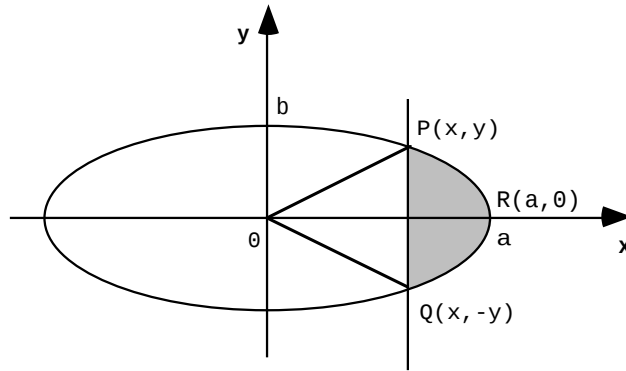
4.4.3. Espacio de trabajo común F_{coord}

Para operaciones con cambio de manos y para tareas donde ambos brazos agarran al mismo tiempo el objeto, el espacio de trabajo asequible por ambos robots es una medida importante. Considerando el corte xy se calcula simultáneamente como la intersección de los planos de trabajo de cada robot [v.fig.].



Representación del espacio de trabajo común F_{coord}

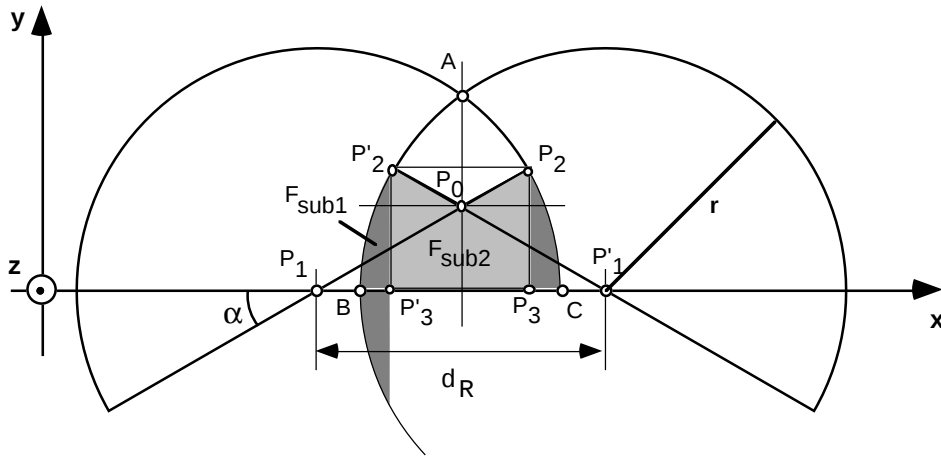
Matemáticamente se puede usar para el cálculo de este plano de intersección, la fórmula para calcular sectores de una elipse.



Cálculo del sector de una elipse

$$\text{área } S_{PQR} = ab \cdot \arccos\left(\frac{x}{a}\right) - xy$$

Puesto que el círculo es un caso especial de un elipse($a=b$), las fórmulas y las propiedades pueden ser utilizadas. Considerando además ciertas relaciones geométricas especiales[v.fig], se puede establecer para el plano la siguiente formula dependiente de d_R y α .



Cálculo del área F_{coord}

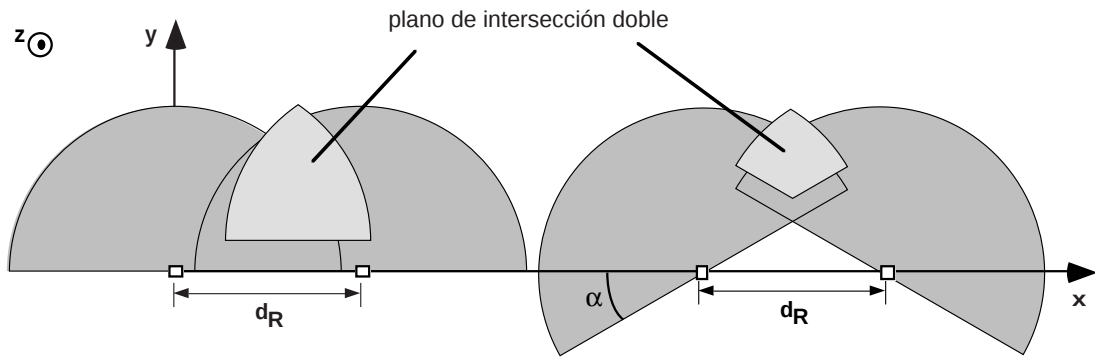
$$F_{coord}(d_R, \alpha) = S_{ABC} - F_{sub1} - F_{sub2} = (r^2 \cdot \arccos\left(\frac{d_R}{2r}\right) - \frac{d_R}{2} \cdot r_{coord}) - F_{sub1} - F_{sub2}$$

$$F_{sub1}(\alpha) = 2(S_{P_1P_2C} - \Delta_{P_1P_2P_3}) = r^2 \cdot (\alpha - \sin \alpha \cdot \cos \alpha)$$

$$F_{sub2}(d_R, \alpha) = S_{P_2P'_2P_3P'_3} - \Delta_{P_2P'_2P_0} = 0,5 \cdot (2r \cos \alpha - d_R)^2 \cdot \tan \alpha$$

4.4.4. Espacio de trabajo total F_{tot}

La medida más evidente para maximizar, es el espacio de trabajo total. El determina esencialmente, que tan flexible, rápido y eficaz se pueden realizar los trabajos (p.e. sin cambiar la posición de la grúa). Se calcula sumando las áreas alcanzables de los robots y substrayendo el área de intersección [v.fig.]. Esta ya fue calculada (F_{coord}) y se puede utilizarlo aquí directamente.



Representación del espacio de trabajo total F_{tot}

En razón de las relaciones geométricas (ambos planos se suman al plano de un círculo) resultando la dependencia siguiente:

$$F_{tot}(d_R, \alpha) = F_{circulo} - F_{coord}(d_R, \alpha)$$

$$\Rightarrow F_{tot}(d_R, \alpha) = \pi \cdot r^2 - F_{coord}(d_R, \alpha)$$

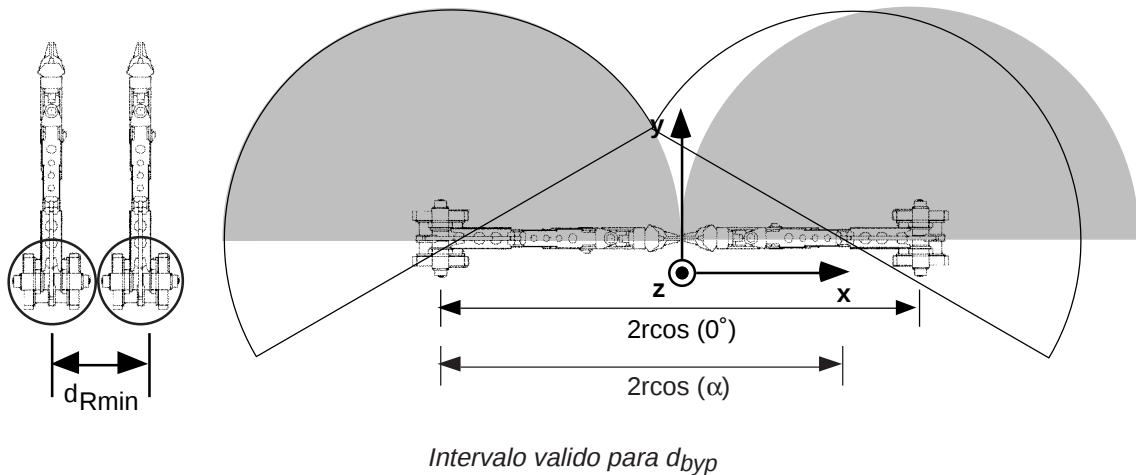
4.5. Clasificación de las medidas encontradas

En este punto se hace una retrospectiva de las medidas encontradas y un primer intento clasificar sus propiedades. Se han encontrado respectivamente dos medidas unidimensionales y dos bidimensionales, las que, en principio sus efectos no sean completamente independiente. Un r_{coord} grande implica una área grande F_{coord} y sucede algo similar con las medidas d_{byp} y F_{tot} . Por tanto se puede proponer que existen respectivamente dos pares de medidas que influyen de forma semejante d_R y a tendencioso en la misma dirección. r_{coord} y F_{coord} acentúan la coordinación de los dos robots (d_R y a pequeño), mientras d_{byp} y F_{tot} pesan mas la independencia de los robots (d_R y a grande). Las dimensiones geométricas de los objetos, manejadas con dos brazos y en una distancia máxima, son consideradas por las

medidas unidimensionales r_{coord} y d_{byp} , mientras las dos medidas de superficie F_{tot} y F_{coord} representan una medida de la flexibilidad y la capacidad global. Se ha encontrado un método de descripción, que deja suficiente libertad para la integración de distintas técnicas de trabajo, pero que, al mismo tiempo, hace posible una expresión general y cuantitativa de las características del presente sistema con dos brazos.

4.6. Distancia mínima y máxima del sistema

Porque la mecánica de los robots y el hecho, que los robots deben trabajar coordinadamente, resultan intervalos, fuera de las contemplaciones anteriores no tienen sentido y además, en el caso del cálculo de los segmentos de elipse, no producen resultados. La distancia mínima resulta de la estructura física de los robots y se denomina d_{Rmin} . Si d_R esta por debajo de este mínimo los robots colisionan. La distancia máxima es el caso límite, en el cual ambos robots apenas pueden tocarse ($d_{Rmax}=2r \cos a$). El intervalo de valores de a se fija a $D_a=[0, \pi/8]$. Valores mayores de a no son oportunos, pues entonces el espacio de trabajo común se disminuye demasiado a costa del espacio lateral en dirección x .



4.7. Distancia y ángulo óptimo considerando todas las medidas

4.7.1. Unificación de las medidas de valoración

El objetivo de este apartado es encontrar la distancia y el ángulo entre los robots, que las cuadros medidas de valoración sean tan grandes como sea posible. Para representar claramente todas medidas en el mismo sistema de referencia (Valores de la función en el intervalo 0..1) es necesario referirlas respectivamente a sus máximos. Las medidas normalizadas son:

$$d_{bypnorm}(d_R) = \frac{1}{C_{ndbyp}} \cdot 2 \cdot d_R$$

$$r_{coordnorm}(d_R) = \frac{1}{C_{nrcoord}} \sqrt{r_R^2 - (d_R/2)^2}$$

$$F_{coordnorm}(d_R, \alpha) = \frac{1}{C_{nFcoord}} \left(r^2 \cdot \arccos\left(\frac{d_R}{2r}\right) - \frac{d_R}{2} \cdot r_{coord} - r^2 \cdot (\alpha - \sin \alpha \cdot \cos \alpha) - 0,5 \cdot (2r \cos \alpha - d_R)^2 \cdot \tan \alpha \right)$$

$$F_{totnorm}(d_R, \alpha) = \frac{\pi r^2 - F_{coord}}{\pi r^2} = 1 - \frac{F_{coord}}{\pi r^2}$$

con

$$C_{ndbyp} = d_{byp}(d_{r \max})$$

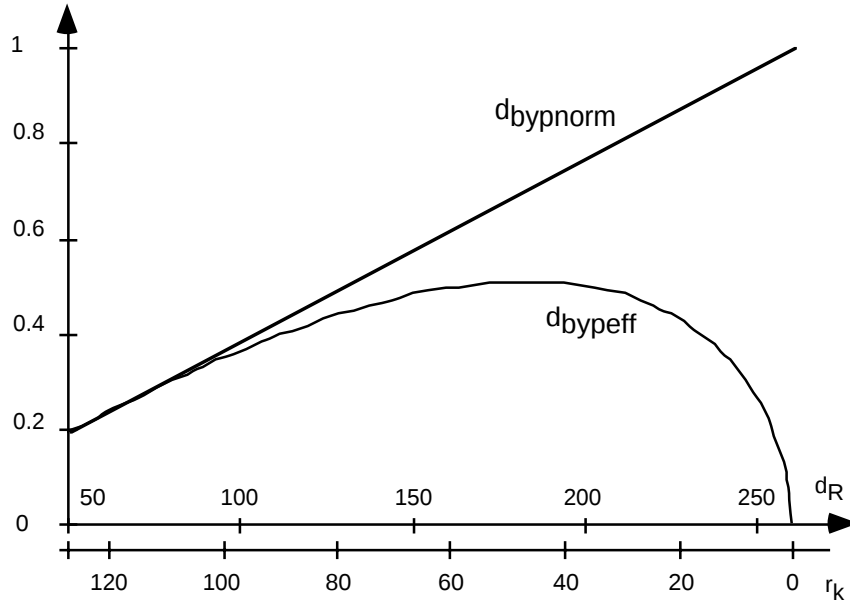
$$C_{nrcoord} = r_{coord}(d_{r \min})$$

$$C_{nFcoord} = F_{coord}(d_{r \min}, \alpha_{\min})$$

La medida $d_{bypnorm}$ posee una dependencia lineal de d_R . Pero en realidad seguramente no tiene su valor óptimo cualitativo en $d_R = 2r$, porque esta medida no es muy representativa cuanto r_{coord} se aproxima mas a 0 (manipulaciones con objetos mas grandes no tienen sentido por la situación mecánica y la localización). Para tener esto en cuenta se multiplica la función por $r_{coordnorm}$. Además esto tiene sentido, porque d_{byp} esta definido directamente respecto a r_{coord} . Por eso resulta la formula de d_{bypeff} , es decir la medida usada para la optimización, como sigue:

$$d_{bypeff}(d_R) = \frac{1}{C_{ndbyp}} \cdot 2 \cdot d_R \cdot r_{coordnorm}$$

La siguiente figura muestra d_{byp} antes y después de la multiplicación por r_{coord} . La división adicional del eje x aclara la disminución fuerte de $d_{bypnorm}$ cerca de $r_{coord}=0$.



Representación gráfica de $d_{bypnorm}$ y d_{bypeff}

4.7.2. Visualización gráfica

La visualización de las cuatro medidas en función de d_R y α , como así la determinación de d_{Ropt} y α_{opt} , se hace con MAPLE [código en el apéndice A.2], un lenguaje para solucionar problemas matemáticos simbólica o numericamente. Todas las medidas de valoración son presentadas en el apéndice A.1 como plots tridimensionales.

4.7.3. Cálculo de la distancia óptima y del ángulo óptimo

Ahora hay que encontrar la distancia y el ángulo, a las cuales todas las medidas sean tan grandes como sea posible. Esto se consigue con una reunión via multiplicación de todas medidas en una función, de la cual se busca su valor extremo. Ya que los valores de las funciones se mueven solo en el intervalo 0..1 y

además el máximo siempre está en el valor óptimo, se define una función de evaluación como sigue:

$$\text{val}(d_R, \alpha) = d_{\text{byeff}}^{k_1}(d_R) \cdot r_{\text{coordnorm}}^{k_2}(d_R) \cdot F_{\text{totnorm}}^{k_3}(d_R, \alpha) \cdot F_{\text{coordnorm}}^{k_4}(d_R, \alpha)$$

4.7.3.1. Pesando las medidas de valoración

En su forma actual, cada medida tiene la misma 'importancia' en el calculo del óptimo. Sin embargo es posible que las acciones esten distribuidas tan heterogéneamente, que tendría sentido pesar algunas medidas mas que las otras. Por ejemplo, si la tarea mas importante es fijar o verificar conexiones, hay que pesar mas la medida F_{coord} , que significa acciones coordinadas (sobre objetos pequeños) con los dos brazos. Para formularlo matemáticamente, se introducen los exponentes $k_1 \dots k_4$. Variando estos exponentes se pueden adaptar las ecuaciones a tareas distintas. Pero esto tiene solo sentido, cuando se puede variar la distancia d_R y el ángulo α directamente in situ. Con un montaje único fijado con acciones no definidas exactamente es mejor dejar los exponentes a 1, para garantizar una configuración tan flexible como sea posible. La ecuación general de optimización será:

$$\text{val}(d_R, \alpha) = d_{\text{byeff}}^{k_1}(d_R) \cdot r_{\text{coordnorm}}^{k_2}(d_R) \cdot F_{\text{totnorm}}^{k_3}(d_R, \alpha) \cdot F_{\text{coordnorm}}^{k_4}(d_R, \alpha)$$

Se puede determinar el máximo de esta función o gráficamente o calcularlo mediante:

$$\text{opt}(d_R, \alpha) = \frac{\partial}{\partial(d_R, \alpha)} (\text{val}(d_R, \alpha)) = 0 \quad ; d_R = d_{R \min} \dots d_{R \max}$$

$$\alpha = \alpha_{\min} \dots \alpha_{\max}$$

Hay que tener en cuenta que funciones con varias variables y con un área de definición restringida tienen dos clases de máximos:

- máximos relativos, que son calculados con la primera derivación de la función
- valores extremos sobre el borde. El borde de un área de definición de n dimensiones es un área con $(n-1)$ dimensiones, cuyos máximos os pueden ser divididos de nuevo en máximos os relativos y valores extremos

Así se puede calcular iterativamente todos los candidatos posibles y se obtiene el resultado por comprobación de valores.

La posibilidad para la determinación de los máximos que se ha descrito anteriormente, en general no es realizable a causa de la complejidad de las funciones que se producen. Con mucha frecuencia tampoco se consigue una solución numérica, porque los métodos iterativos no convergen.

Por eso se recomienda aquí una manera de proceder pragmática determinando el valor de α en el máximo y transformando la función bidimensional en unidimensional. Se puede derivar fácilmente la función resultado y se obtiene d_R solucionando la ecuación

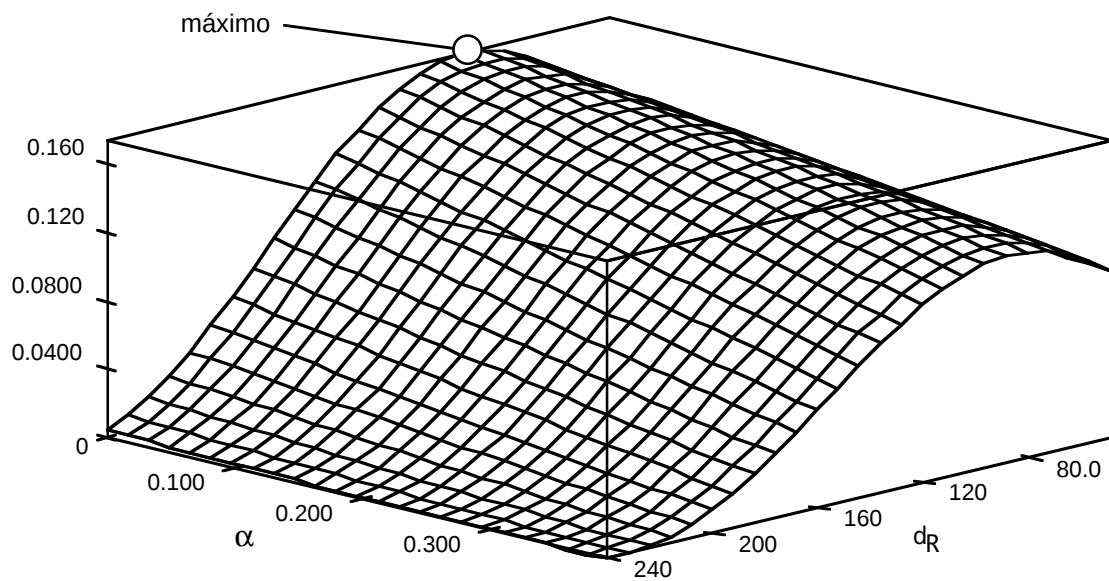
$$\text{opt}(d_R, \alpha) = \frac{\partial}{\partial d_R} (\text{val}(d_R, \alpha = \text{const.})) = 0$$

En el apéndice A.3 se como se obtiene el valor de α , la función unidimensional y la derivación correspondientes para los dos casos $k_3=1$ y $k_3=8$. Se puede reconocer muy bien que solo existe un máximo con $\alpha \neq 0$ con un peso muy grande en el espacio total ($k_3 \geq 7$). Pero incluso en el caso $\alpha \neq 0$ los valores de la función son muy similares al del máximo, así que también se puede aceptar $\alpha=0$ sin modificar mucho el resultado.

En la visualización de la función $\text{val}(d_R, \alpha)$ para $k_i = 1$; $i = 1 \dots 4$ se reconoce muy bien el máximo en $\alpha=0$ y $d_R \approx 100\text{cm}$.

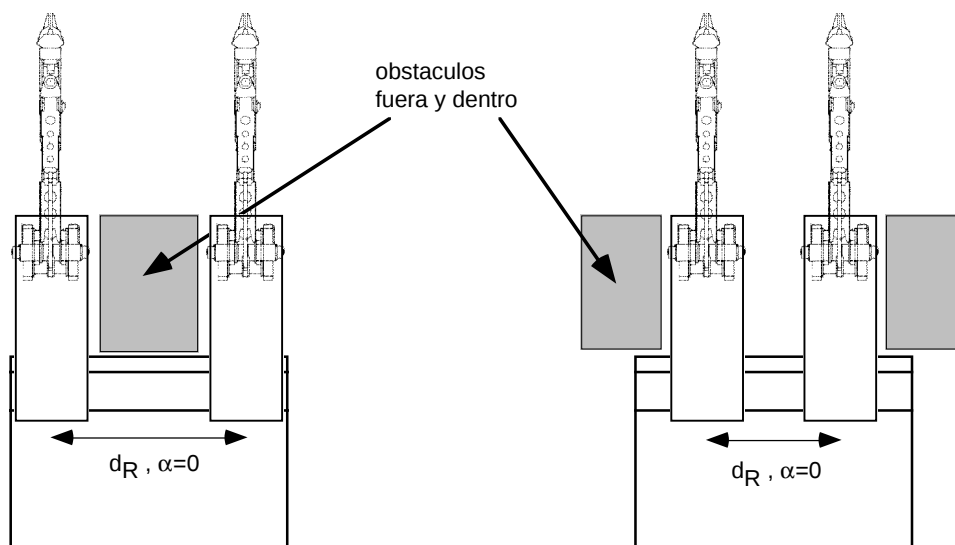
El valor exacto con $r=130\text{cm}$ es

$$\alpha = 0 \quad ; \quad d_{R\text{opt}} = 104,2\text{cm}$$

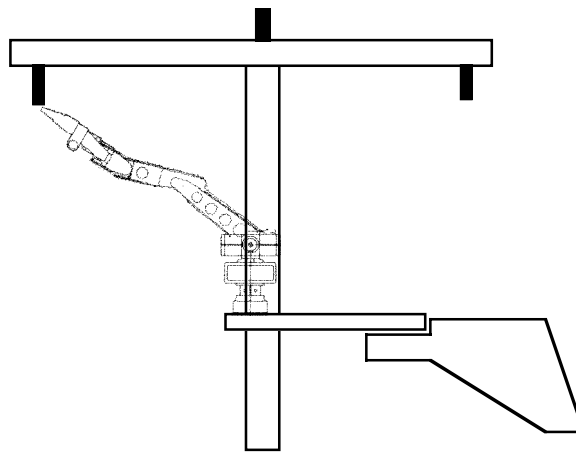


Representación tridimensional de la función de calidad de la disposición

Ya que la función no varía mucho alrededor del máximo, sería posible considerar una adaptación para responder a ciertos aspectos especiales. Por ejemplo un montaje con un d_R variable, para evitar obstáculos [v.fig.].



Vista plano xy



Vista plano yz

En la tabla siguiente se demuestra la influencia de los exponentes sobre d_R . No se contempla aquí el ángulo α , porque las investigaciones mostraron que solo con un peso muy alto en la medida correspondiente al espacio total F_{tot} , resulta un α distinto a 0. Las medidas de valoración están dadas para d_{Ropt} en $\alpha=0$ y $k_i = 1$; $i = 1...4$, en el intervalo ± 10 cm en pasos de 2cm, para mostrar las variaciones cerca del optimum. La medida subrayada está pesada con $k_i = 2$. Las longitudes están dadas en cm y las superficies en metros cuadrados.

comentario	k ₁	k ₂	k ₃	k ₄	d _R	r _{coord}	d _{byp}	F _{coord}	F _{tot}
peso proporcionado	1	1	1	1	94,27	121,2	188,5	1,46	3,85
					96,27	120,8	192,5	1,43	3,88
					98,27	120,4	196,5	1,41	3,9
					100,3	119,9	200,5	1,38	3,93
					102,3	119,5	204,5	1,36	3,95
optimo calculado	1	1	1	1	104,3	119,1	208,5	1,34	3,97
					106,3	118,6	212,5	1,31	4
					108,3	118,2	216,5	1,29	4,02
					110,3	117,7	220,5	1,27	4,04
					112,3	117,3	224,5	1,24	4,07
					114,3	116,8	228,5	1,22	4,09
mayor peso d _{byp}	2	1	1	1	128,4	113	256,8	1,06	4,25
mayor peso r _{coord}	1	2	1	1	96,76	120,7	193,5	1,43	3,88
mayor peso F _{tot}	1	1	2	1	117,2	116	234,5	1,18	4,13
mayor peso F _{coord}	1	1	1	2	73,93	124,6	147,9	1,71	3,6
mayor peso r _{coord} y F _{coord}	1	2	1	2	70,55	125,1	141,1	1,75	3,56

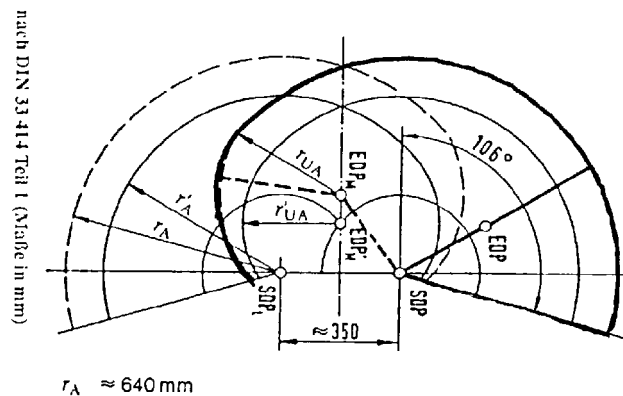
Se muestra claramente como los exponentes k_i influyen respectivamente en la medida pesada, y también, que tanto d_{byp} y F_{tot} como r_{coord} y F_{coord} son influidas simultáneamente en la misma dirección.

4.8. Comparación con la proporción espalda/brazo del hombre

Una comprobación con las medidas y conocimientos de la antropometría parece interesante por las razones siguientes:

- El diseño del sistema con 2 robots responde al manejo intuitivo y la ergonomía del sistema es similar al manejo manual.
- La cinemática de los robots es parecido al brazo del hombre.
- La colocación de los robots encima de la plataforma es análogo al sistema 'brazo/espalda'.

La figura muestra algunas medidas típicas para un hombre (5. percentil).



Medidas típicas del hombre (5. percentil)

La proporción de la anchura de espalda a longitud de brazo asciende a 0,546. Según esto la distancia de los robots tuviera que ser 70,98cm. La cuestión que aquí nos interesa es cuáles valores de k_j corresponden a la situación humana. Lo que marca la capacidad del hombre para manejar cosas son seguramente sus acciones coordinadas. Contemplando por eso la función en la tabla con el peso mayor en r_{coord} y F_{coord} , que representan la cooperación, se encuentra una distancia de 70,55cm. Naturalmente no es posible hacer una comprobación exacta, debido a las simplificaciones, las relaciones de las tareas particulares y las diferencias entre el sistema técnico y el hombre. Sin embargo parece que esta comprobación confirma tendencialmente nuestros resultados.

4.9. Conclusiones y posibilidad de una ampliación

Se ha encontrado un método para determinar la disposición de dos robots sobre una plataforma mediante una solución analítica. En razón de la complejidad se han reducido las contemplaciones tridimensionales necesarias a bidimensionales. Si se quiere utilizar el sistema también in situ para variar interactivamente los parámetros de la disposición de los robots, la velocidad de los cálculos es una condición importante. Por eso hay que aceptar las inexactitudes resultadas.

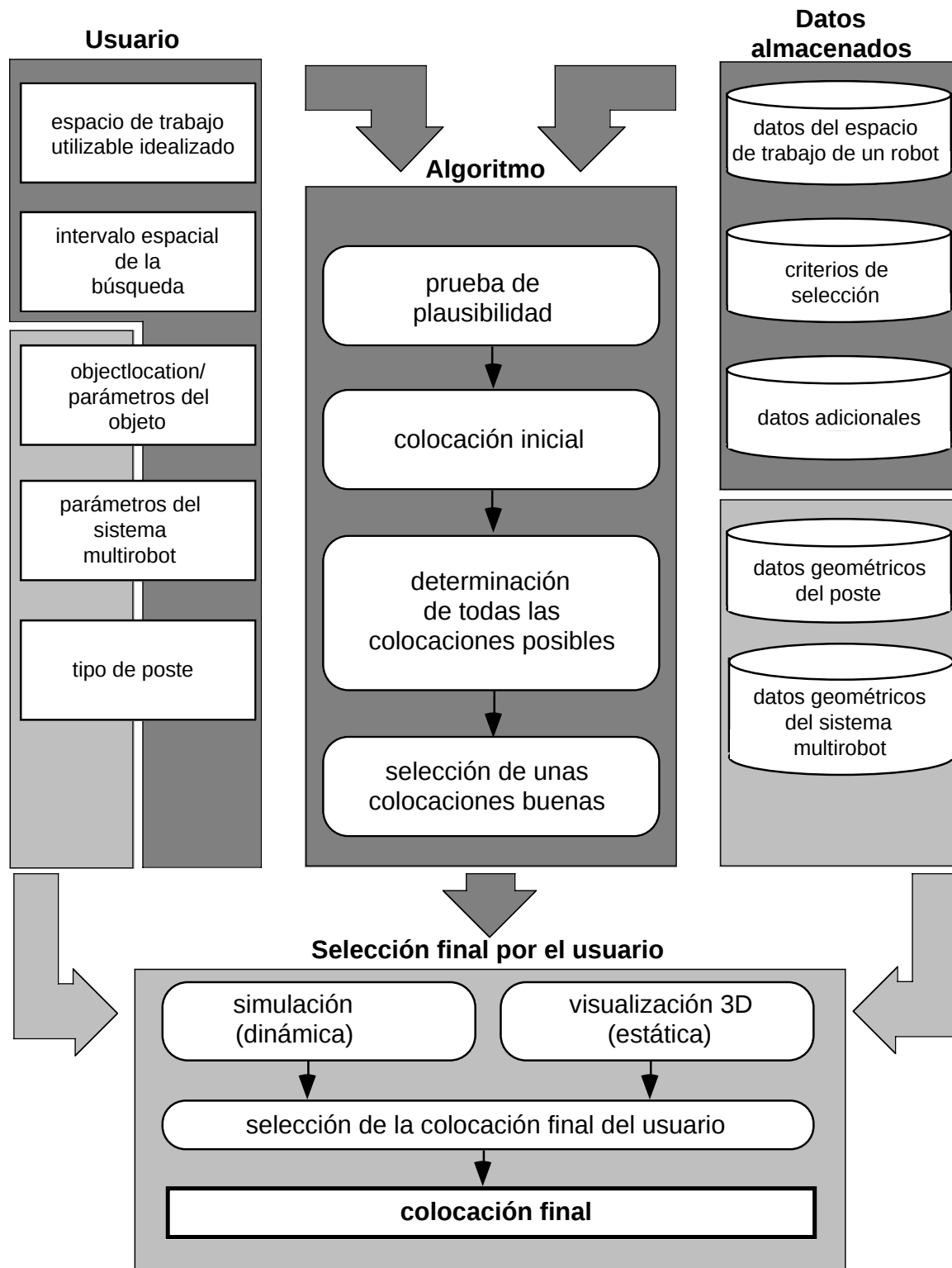
Una ampliación del método sería la formulación de más medidas características o su cálculo más exacto, por ejemplo, tridimensionalmente utilizando métodos numéricos.

5. Algoritmos para la colocación asistida por computador de un sistema multirobot frente a un poste

Para la colocación de un sistema robotizado en un entorno industrial fijo, sin importar si se trata de un único robot o un sistema multirobot, se tiene la posibilidad de diseñar el abastecimiento de los objetos manipulados paralelamente con el posicionamiento del robot. Como además el número de las manipulaciones distintas es en general muy limitado, con mucha frecuencia se puede decidir fácilmente sobre una colocación determinada. Pero en nuestro caso el entorno ya está definido y la adaptación tiene que realizarse sólo del lado del sistema del robots. Por la multitud de trabajos con muchos tipos de postes es necesario, para una actuación eficaz, proyectar previamente la colocación apropiada para las acciones. Con este procedimiento, no solo resulta una realización más rápida in situ, sino que se evita también 'probar' colocaciones moviendo la grúa, lo que lleva fácilmente a un deterioro y provoca un riesgo para los operarios. Sin embargo, esto no significa que no se consideren experiencias anteriores del trabajo manual. Más bien depende de una planificación buena, de una entrada de datos inteligente y de una selección final versátil. Esta herramienta debe dispensar al usuario de la realización de experimentos molestos y darle un modelo gráfico de su entorno de trabajo con todos los componentes importantes. Para este trabajo se usa el simulador para robots TOROS [9].

La siguiente figura muestra la estructura del software. A la izquierda se describen los datos introducidos directamente por el usuario, mientras que a la derecha se encuentran los datos que fueron calculados apriori como descripción del espacio de trabajo o como criterios de selección. La descripción del espacio de trabajo ya se encuentra en un formato apropiado en disco duro, porque un cálculo dentro del algoritmo costaría demasiado tiempo. Se resuelven los problemas de cálculos intensivos antes y se almacenan una multitud de soluciones para elegir la correspondiente en cada caso particular .

El algoritmo ofrece al usuario un número manejable de colocaciones buenas, de las cuales puede elegir la colocación final usando una simulación (apéndice B.10) o simplemente una visualización tridimensional.



Estructura del software

5.1. Suposiciones y simplificaciones

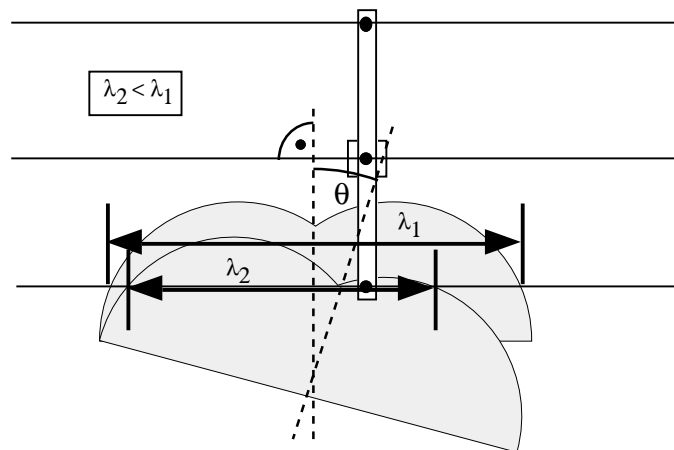
5.1.1. Orientación del poste

Para simplificar la sistematización del desarrollo de los algoritmos, se define la orientación de los postes, tal que las líneas estén paralelas al eje x del sistema de referencia del mundo [v.fig].

5.1.2. Orientación de los robots respecto al poste

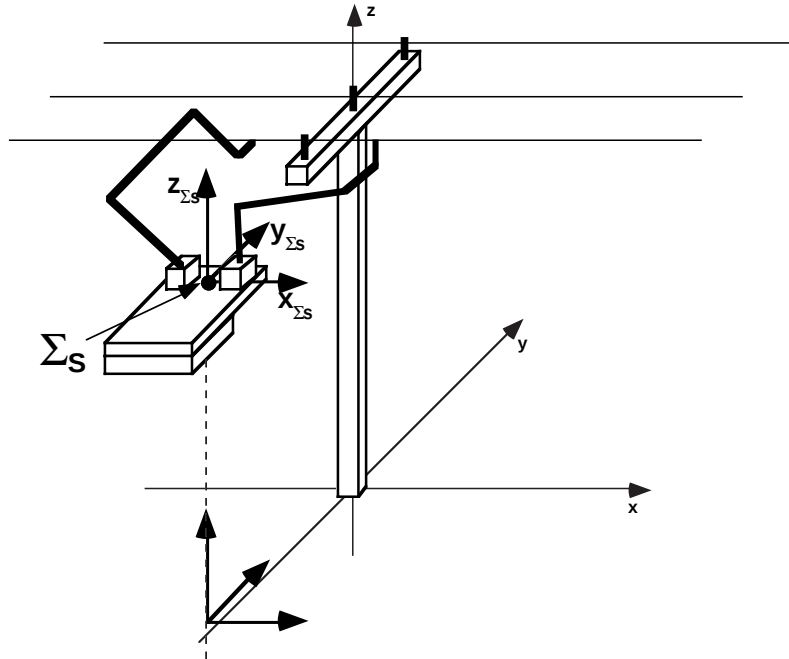
Aquí parece que tiene sentido limitar también la totalidad de las colocaciones posibles del sistema respecto al poste, para reducir la complejidad. Para ello se hace algunas reflexiones:

- Un posicionamiento del sistema tal que la orientación de los brazos coincide con la dirección de las líneas tiene poco sentido, porque las líneas solo permitirían una colocación por debajo de ellas. Esto reduciría el espacio de trabajo utilizable a un mínimo y sería además muy difícil el acceso de la grúa. Como posibilidad práctica quedaría una colocación al lado de las líneas, que tampoco parece muy buena.
- Por otra parte la disposición del sistema vertical a las líneas ofrece un acceso a los objetos bastante flexible. Cada desviación de esta posición vertical reduce el alcance del sistema multirobot[v.fig].



Disposición vertical del sistema frente al poste

De estas reflexiones resulta el escenario presentando en la figura siguiente, que se establece como supuesto de la discusión. El sistema de referencia del sistema robotizado Σ_S es colocado exactamente en el punto medio de los dos sistemas de referencia de los robots (Σ_L y Σ_R) con la misma orientación.



Orientación del sistema relativa al poste

5.1.3. Ningún cambio de la colocación durante las manipulaciones

Además, se define que sólo se pueden planificar las acciones que no requieren un cambio de la colocación mientras se están realizando. Procesos más complejos y fuera del entorno de trabajo inicial tienen que ser divididos en más simples, que se ejecutan secuencialmente con el cambio de posición correspondiente. Estas *subacciones* únicas son planificadas tal como si fueran acciones completas.

5.2. Nomenclatura

5.2.1. Conceptos

espacio de trabajo

asequible Espacio de trabajo que incluye todos los puntos que pueden ser alcanzados independientemente de la orientación

espacio de trabajo

práctico Espacio de trabajo que incluye todos los puntos que pueden ser alcanzados en cualquier orientación

espacio de trabajo

utilizable idealizado Esa parte del espacio de trabajo del robot que se considera por el algoritmo de posicionamiento. La definición procede de un espacio rectangular con las medidas h_{IAR} , b_{IAR} y t_{IAR} relativas al punto P_{IAR} .

espacio de trabajo utilizable intersección del espacio de trabajo real con el espacio de trabajo utilizable idealizado

frame ver frame de la pinza

frame de la pinza matriz 4x4, que contiene la información de la posición, orientación.

location posición y vector $\vec{a}$ que representa una unidad de volumen discreta del espacio

objectlocation..... posición y orientación del agarre de un objeto en forma de un punto (posición y vector $\vec{a}$)

EAO Objeto que es manipulado con un brazo

ZAO Objeto que es manipulado coordinadamente con dos brazos

COI Código del intervalo de orientación

COT Código del tipo de objeto (EAO, ZAO)

5.2.2. Símbolos

d	resolución del intervalo espacial de la búsqueda
$(\vec{n}, \vec{o}, \vec{a})$	sistema de referencia de la pinza con los vectores $\vec{n}$ (<i>normal</i>) $\vec{o}$ (<i>orientation</i>) $\vec{a}$ (<i>approach</i>)
n_{INT}	número de los intervalos espaciales de la búsqueda
n_S	número der posiciones discretas del intervalo espacial de búsqueda definido por el operario
n_{aut}	número de posiciones discretas del intervalo espacial óptimo de búsqueda calculado por el programa
n_V	número de las unidades de volumen en los que se divide el espacio de trabajo
res	resolución de los espacios de trabajo almacenados
P_{IAR}	punto de referencia para la definición del espacio de trabajo utilizable idealizado
$b_{IAR}, h_{IAR}, t_{IAR}$	ancho, altitud y profundidad del espacio de trabajo utilizable idealizado
v_x, v_y, v_z	dimensiones en x,y y z del intervalo de búsqueda en la unidad d
Db, Dh, Dt	dimensión del paralelepipedo rectangular que circunscribe todos los objetos (EAOs y ZAOs)
$Db_{coord}, Dh_{coord}, Dt_{coord}$	dimensión del paralelepípedo rectangular que circunscribe todos los objetos manipulados coordinadamente (ZAOs)
S_O, S_{AR}	centro de gravedad de los objetos y del espacio de trabajo
S_L, S_R	sistemas de referencia del robot a la izquierda y a la derecha
S_S	sistema de referencia del sistema multirobot entero
S_K	sistema de referencia de la grúa
R_L, R_R	Símbolo para el robot de la izquierda y de la derecha

t_x, t_y, t_z	vector de traslación de un objeto (parámetros adicionales del objeto)
r_j, r_J	ángulo que define la rotación de un objeto (parámetros adicionales del objeto)

5.3. Datos necesarios

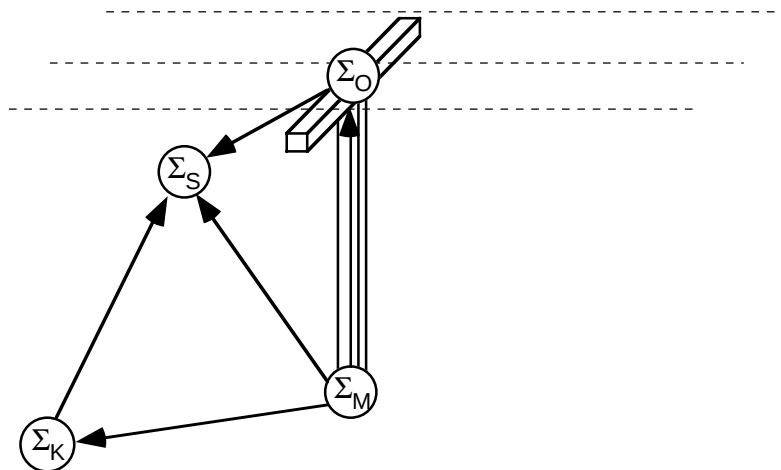
En los siguientes apartados se explican las medidas más importantes para el algoritmo. Algunos de estos datos se encuentran en un formato apropiado en disco duro, mientras que otros tienen que ser introducidos por usuario para cada planificación.

5.3.1. *Tipo del poste*

Es claro que el poste o más exactamente la geometría del poste es muy importante para la planificación. Los datos para la determinación de una colocación apropiada, incluyendo una simulación o visualización, se pueden dividir en tres clases. Para almacenarlos se propone una base de datos, con la que se tiene un acceso muy fácil a los distintos postes y que puede ser actualizado con nuevas geometrías.

5.3.1.1. *Sistemas de referencia*

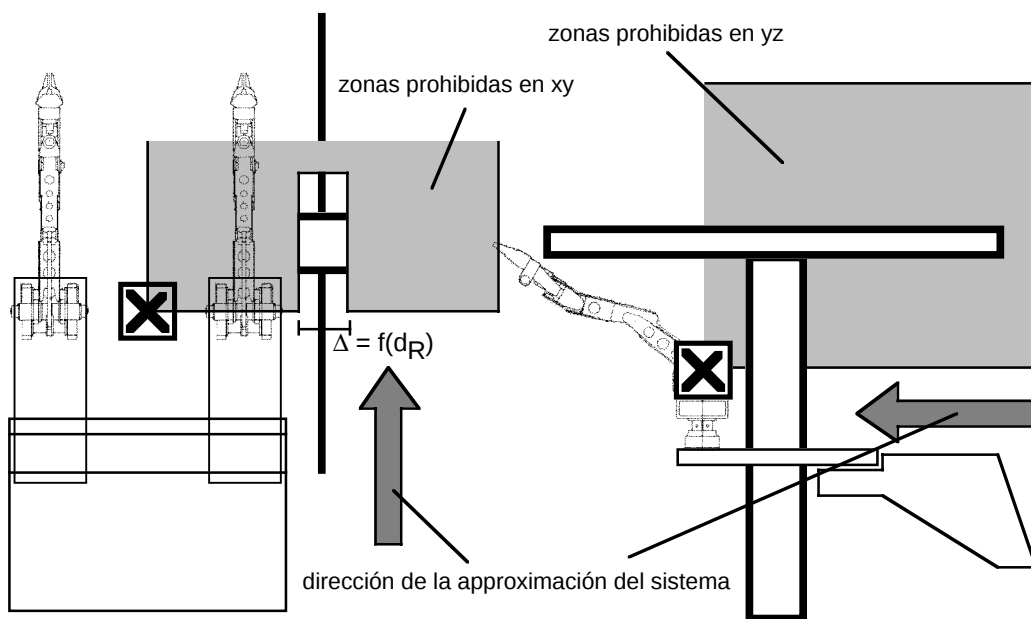
El poste tiene la base fija en el entorno, respecto a la cual es definido el sistema y todos los demás objetos. En general, se pone la base S_M en el punto de pie del poste con la orientación descrita. Adicionalmente se puede definir un segundo sistema de referencia S_O para simplificar la definición de los objetos, que resulta de una traslación de S_M y que considera la geometría especial del poste. Ya que los sistemas de referencias S_M , S_O y S_S tienen, de acuerdo con las suposiciones, la misma orientación, resultan las siguientes relaciones con la base de la grúa S_K .



Relación entre varios sistemas de referencia

5.3.1.2. Zonas prohibidas para la colocación

Debido a que a veces los postes tienen una mecánica compleja y teniendo en cuenta el tamaño de la plataforma, habrá zonas en las que un posicionamiento del sistema sería peligroso (distancia de seguridad) o imposible (colisiones). Se describen estas zonas de una manera adecuada relativas a Σ_M y se utilizan luego para los criterios de selección de las soluciones posibles. Estas áreas del espacio dependen también de la geometría de la plataforma. Por eso hay que definirlas dependiendo de los parámetros variables del sistema (por ejemplo d_R)



Zonas prohibidas

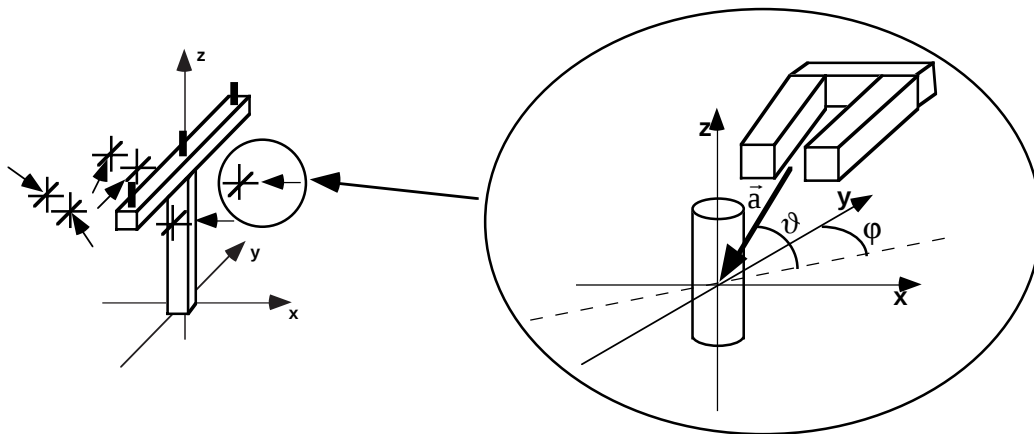
No se trata aquí el problema de si una descripción en dos vistas tiene sentido o si se usa mejor una descripción tridimensional, porque no es muy importante para la comprensión del algoritmo.

5.3.1.3. Datos geométricos del poste

Para una simulación o una visualización se necesitan las medidas geométricas del poste. Estos datos no son exigidos por el algoritmo, porque toda la información importante para la colocación ya está dentro de la descripción de las zonas prohibidas y de los sistemas de referencia.

5.3.2. Posición y orientación del objeto

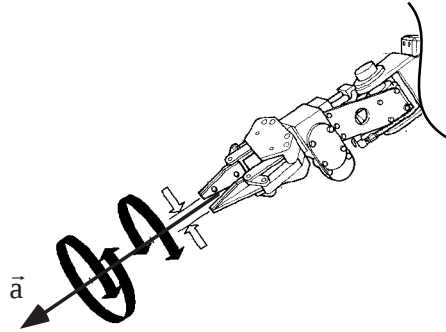
Primero suponemos que no importan las manipulaciones particulares para el posicionamiento de los robots, sino sólo donde se encuentran y cuál es la dirección en la que se los agarra. Esta dirección está representada por la orientación del sistema de referencia del objeto, porque para agarrarlo tiene que coincidir con el sistema de referencia de la pinza. Se define entonces para cada objeto un *objectlocation*, que se compone de su posición y orientación. No se contempla la dimensión de los objetos, porque ya se encuentran descritos completamente por un punto y su orientación [v.fig].



Descripción de los objetos como 'objectlocation'

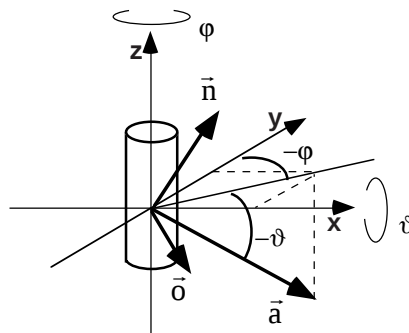
Para la determinación de la orientación es suficiente, debido a la mecánica especial de la pinza, especificar sólo el vector $\vec{a}$ (vector en la dirección de la aproximación, *approach*). El robot GRIPS es capaz de girar continuamente su pinza sobre el vector

$\vec{a}$ (*continuous mode*). Se verá también luego, mediante las ecuaciones cinemáticas inversas, que se necesitan sólo los parámetros de la orientación a_x , a_y , a_z y los parámetros de la posición p_x , p_y , p_z para un cálculo del espacio de trabajo.



modo continuo del robot GRIPS

Debido a que el origen del sistema de referencia de la pinza ($\vec{n}, \vec{o}, \vec{a}$) coincide con el del objeto, se puede trasladar para nuestra tarea con dos rotaciones de los ángulos J (sobre eje x) y j (sobre eje z) sin una traslación.



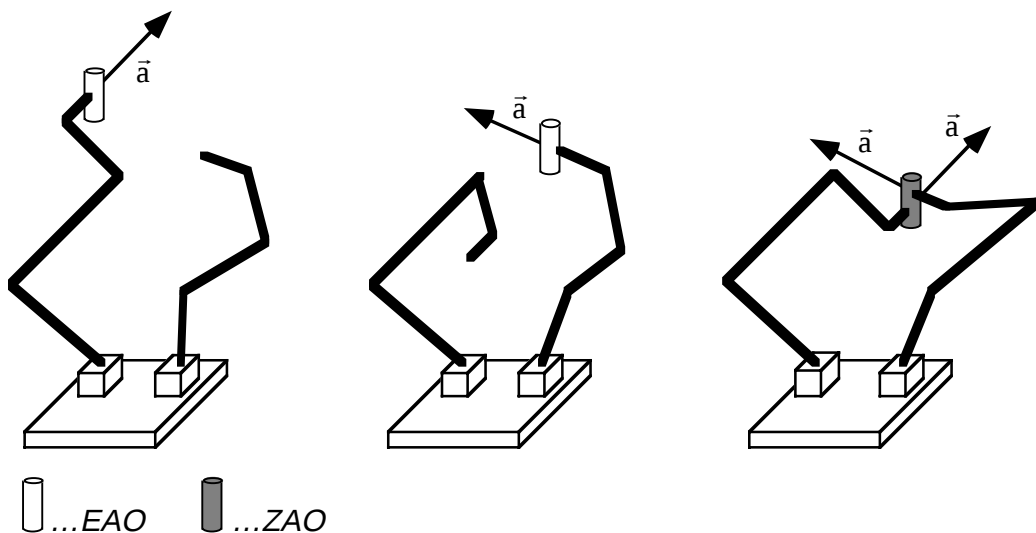
rotación del sistema de referencia de la pinza sobre el eje x y z

Aquí quería señalar que con rotaciones sobre el eje x y el eje z sólo el vector $\vec{a}$ está definido exactamente. Para una descripción exacta de la orientación de los tres vectores ($\vec{n}, \vec{o}, \vec{a}$) se necesitarían tres ángulos .

Cuando se habla entonces de la orientación se piensa en nuestro caso especial en la orientación del vector $\vec{a}$ para agarrar. Esta orientación sólo está determinada por los dos ángulos J y j , aunque para el caso más general se necesitan tres.

5.3.3. Parámetros adicionales del objeto

Debido a que tenemos un sistema con dos robots, es importante para la asequibilidad de un objeto, saber que robot lo va a manipular. Se distingue entonces entre objetos que son manipulados por uno de los robots y objetos manipulados coordinadamente. En la definición de un objeto para una manipulación coordinada se especifica la orientación del agarre separadamente para cada robot. La información, si un objeto es manipulado con uno o dos brazos es muy importante para las reflexiones siguientes, ya que repercute mucho en la colocación, porque el espacio de trabajo común restringe mucho el numero de las colocaciones. Por lo tanto se añade el código del tipo del objeto (EAO o ZAO) y la asignación al robot como parámetro adicional al *objectlocation*.

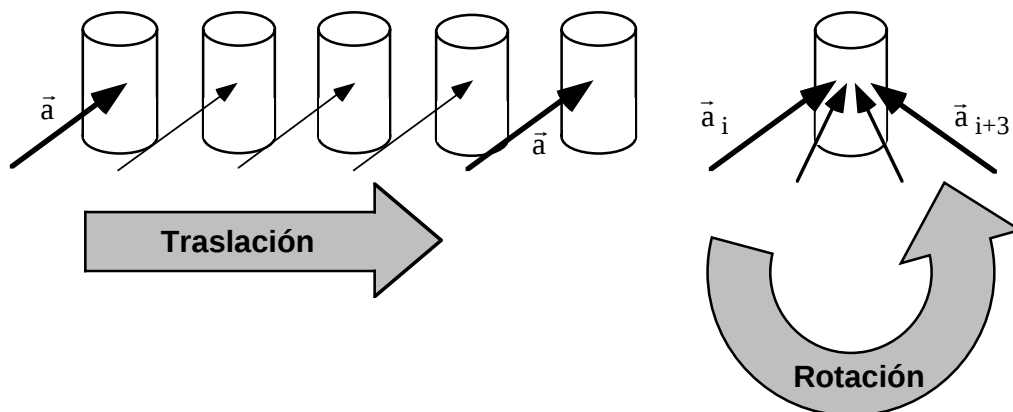


división de los objetos en EAO y ZAO

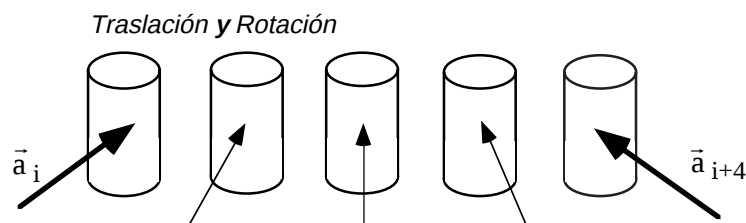
En la tabla siguiente se ven todos las variables de entrada hasta ahora necesarias.

tipo del objeto	Posición			Orient.		Robot	
	x	y	z	j	J	R _L	R _R
COT							
EAO cualquiera	-	-	-	-	-	1	1
EAO derecha	-	-	-	-	-	0	1
EAO izquierda	-	-	-	-	-	1	0
ZAO	-	-	-	-	-	0	1
				-	-	1	0

Alcanzar un objeto no garantiza que se lo puede manipular. Por ejemplo, hay que poder girar o trasladar el objeto. La idea para considerar este problema se basa en la suposición de dividir un giro en un numero finito de pasos, todos con la misma posición pero con una orientación discreta entre la inicial y la final. De la misma forma se realiza la traslación.



Se puede imaginar fácilmente una combinación entre los movimientos básicos tal como en la siguiente figura.



Está claro que la división de un movimiento dinámico continuo en objetos estáticos no es válida en general, porque a veces se presenta un cambio de la configuración entre dos objetos (brazo izquierdo, brazo derecho, codo arriba, codo abajo, ...). Pero

debido a que en nuestro caso sólo se considera una única configuración, este método práctico tiene sentido. Por eso hay que especificar todavía el vector de traslación (t_x , t_y , t_z) y el ángulo de rotación (r_j , r_J).

Si se maneja muy intensivamente un objeto se debe considerar de forma más importante en la colocación de los robots. Para garantizarlo se introduce un peso g_i , que acentúa la importancia del objeto i . La tabla siguiente muestra los objetos posibles con sus parámetros:

COT	x	y	z	t_x	t_y	t_z	j	J	r_j	r_J	g	R_L	R_R
EAO cualquiera	-	-	-	-	-	-	-	-	-	-	-	1	1
EAO	-	-	-	-	-	-	-	-	-	-	-	0	1
EAO	-	-	-	-	-	-	-	-	-	-	-	1	0
ZAO	-	-	-	-	-	-	-	-	-	-	-	0	1
				-	-	-	-	-	-	-		1	0

5.3.4. Datos geométricos del sistema multirobot

Debido a que el sistema se compone de una plataforma de geometría previamente no determinada y dos robots, se tienen que introducir los parámetros que cambian la geometría del sistema. En nuestro caso sólo es la distancia de los robots d_R . Todos los demás datos geométricos importantes para la colocación ya aparecen en otras partes (Zonas prohibidas / Tipo del poste).

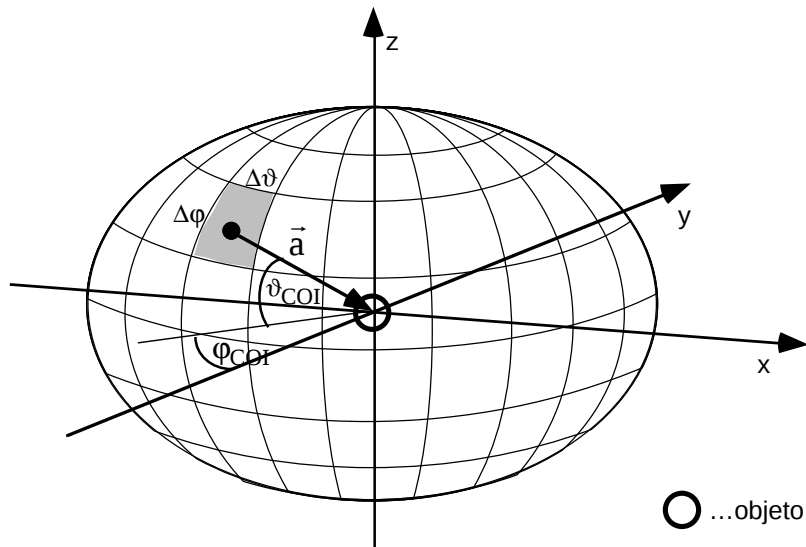
5.3.5. Datos del espacio de trabajo

La descripción del espacio de trabajo que ofrece el fabricante del robot GRIPS, considera sólo el espacio de trabajo asequible. Si se intenta hacer manipulaciones complejas no sirve de mucho tomar un objeto en una orientación cualquiera. Como se deduce también debido a la definición de los objetos por posición y orientación, se tiene que poner el objeto en la parte del espacio que es asequible por el robot con la orientación exigida. Becquet solo distinguía entre espacio de trabajo asequible y práctico. Agarrar un objeto en cualquier orientación o en todas las orientaciones son las dos situaciones extremas de entre todas las posibles. Muchas veces el espacio de trabajo no existe o es pequeñísimo debido a los límites

mecánicos del robot. Se ve entonces que esta división no sirve para nuestro problema de posicionamiento.

Sería necesario una descripción del espacio de trabajo para cada orientación. Una determinación analítica en forma de soluciones relativamente complejas no es posible, porque todos los métodos existentes hacen demasiadas restricciones respecto a la orientación.

Solamente, por tanto, queda la posibilidad de una división discreta orientada al volumen del espacio de trabajo, su cálculo para distintas orientaciones y luego su almacenamiento de una manera apropiada. Para la realización se usa un algoritmo de scanning , que rastrea el espacio según curvas definidas. Por la infinidad de orientaciones posibles no se puede realizar este algoritmo y por eso se divide el espacio en n_{INT} intervalos de orientación (DJ, Dj) , que son representados (y aproximados) respectivamente por una orientación discreta del vector $\vec{a}$.



División del espacio en intervalos de orientación

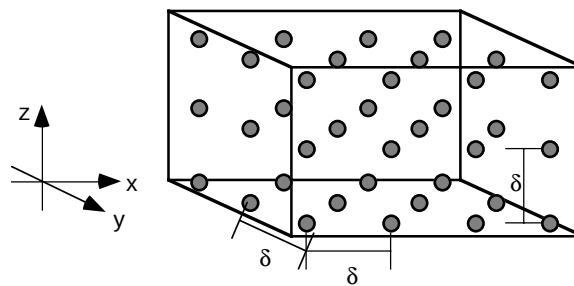
La exactitud de esa aproximación depende naturalmente del tamaño de los intervalos (DJ, Dj) y se tiene que buscar un compromiso justificable entre exactitud y número de las soluciones almacenadas. Se puede reducir el número de los intervalos todavía más si se usan sólo los accesos interesantes para nosotros, es decir con una componente y negativa del vector $\vec{a}$.

$$\begin{aligned} \forall \vec{a} &= (a_x, a_y, a_z) \\ \text{con } a_y &\leq 0 \end{aligned}$$

Estos intervalos de orientación son codificados (COI) para poder relacionar luego fácilmente objeto y espacio de trabajo almacenado.

5.3.6. Determinación del intervalo de la búsqueda espacial

El algoritmo para la colocación busca a partir de un posicionamiento inicial en un intervalo espacial todas las disposiciones que cumplen diversos criterios. Con este fin se representa el espacio como un volumen que circunscribe un numero de puntos discretos equidistantes. La distancia es determinada por la resolución d [v.fig.].



Intervalo de búsqueda representando puntos discretos

Se introduce el intervalo en las direcciones x, y y z por un múltiplo de la resolución d (v_x , v_y , v_z). Entonces resulta, por ejemplo, para el dibujo anterior

$$v_x = 3 ; v_y = 2 ; v_z = 2$$

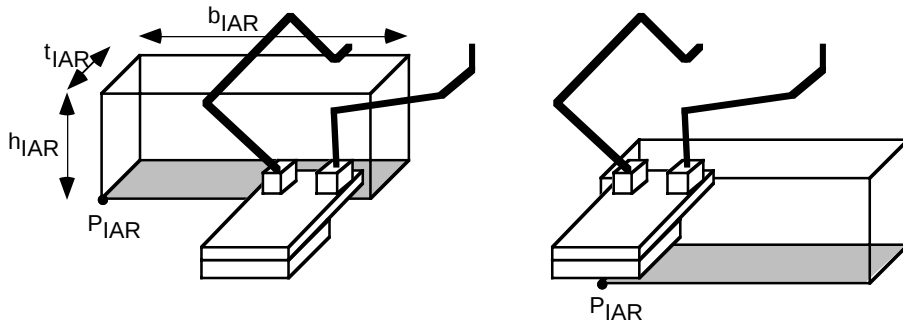
Está claro que con una elección apropiada se puede influir la velocidad del algoritmo considerablemente. La resolución mínima del intervalo de búsqueda d coincide con la resolución res de los espacios de trabajo almacenados y las d más grandes tienen que ser múltiplos de res .

La introducción de este intervalo de búsqueda por el operario no es obligatorio. Tiene también la posibilidad de que el programa calcule automáticamente el intervalo óptimo.

5.3.7. Definición de la parte del espacio de trabajo del robot que se tiene en cuenta

Según la tarea y el lugar de uso del sistema, puede tener sentido sólo marcar alguna parte del espacio de trabajo como utilizable. Esto es necesario, por ejemplo, si se quiere manipular objetos a una distancia mínima o si se quiere excluir partes del

espacio que no se pueden ver muy bien con las cámaras existentes. Se define el espacio de trabajo considerado como intersección del espacio de trabajo total con un cuerpo de volumen en forma de paralelepípedo rectangular (*espacio de trabajo utilizable*). Este paralelepipedo está caracterizado por su punto de referencia P_{IAR} y su altitud h_{IAR} , anchura b_{IAR} y profundidad t_{IAR} y se le llama en lo sucesivo *espacio de trabajo utilizable idealizado*. Cada objeto que se encuentra fuera del espacio de trabajo utilizable pasa entonces a ser no asequible, aunque quizás sería asequible en el espacio de trabajo total. La figura siguiente muestra dos casos del uso de la parte superior izquierda e inferior derecha del espacio de trabajo.



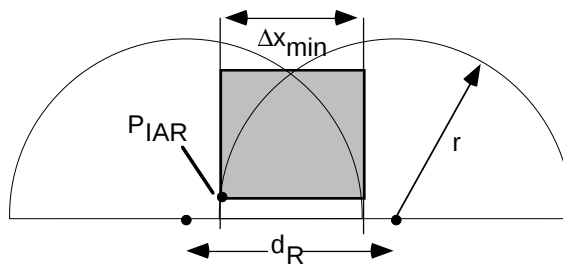
Ejemplo de la definición del espacio de trabajo utilizable idealizado

En razón de la importancia del espacio de trabajo común se prohíbe restringir su dimensión máxima en la dirección x . Por eso el tamaño mínimo del *espacio de trabajo utilizable idealizado* (relativo al punto P_{IAR}) resulta:

$$\Delta x = \Delta x_{\min} = (2r - d_R)$$

$$\Delta y = t_{IAR}$$

$$\Delta z = h_{IAR}$$



Zona no restringible del espacio de trabajo común en la dirección x

Si no se define aquí ninguna restricción, el espacio de trabajo utilizable idealizado coincide con el cuerpo circunscrito en forma de un paralelepípedo rectangular del espacio de trabajo total.

5.4. Descripción del espacio de trabajo considerando orientaciones

5.4.1. Ecuaciones cinemáticas del robot

Casi todos los tipos de robots se componen de una secuencia de articulaciones de traslación y rotación, que son representadas por una cadena cinemática abierta. Teniendo en cuenta los límites de cada articulación resulta el espacio asequible por el robot. Se hace la descripción del espacio de trabajo con la formulación matemática de la cadena cinemática teniendo especialmente en cuenta las orientaciones. Pero la complejidad crece muy rápidamente con el número de los grados de libertad, así que una descripción solo es posible mediante matrices. El método más utilizado es el de Denavit-Hartenberg (método DH). Se basa en asignar a cada articulación un sistema de referencia tal que es posible pasar de un sistema al siguiente usando máximo dos rotaciones y dos traslaciones. Con el uso de coordenadas homogéneas en el espacio euclídeo se puede utilizar matrices 4x4 de transformación, que incluyen rotación, traslación e información de escala y perspectiva. Una matriz homogénea representa entonces la transformación de un vector en coordenadas homogéneas de un sistema de referencia a otro. Una matriz de transformación DH se compone de las siguientes submatrices (transformación de un vector del sistema de referencia n a $n+1$)

$$DH_{n,n+1} = \begin{bmatrix} R & T \\ P & S \end{bmatrix}$$

$R \dots 3 \times 3$ – matriz de rotación

$T \dots 3 \times 1$ – matriz de traslación

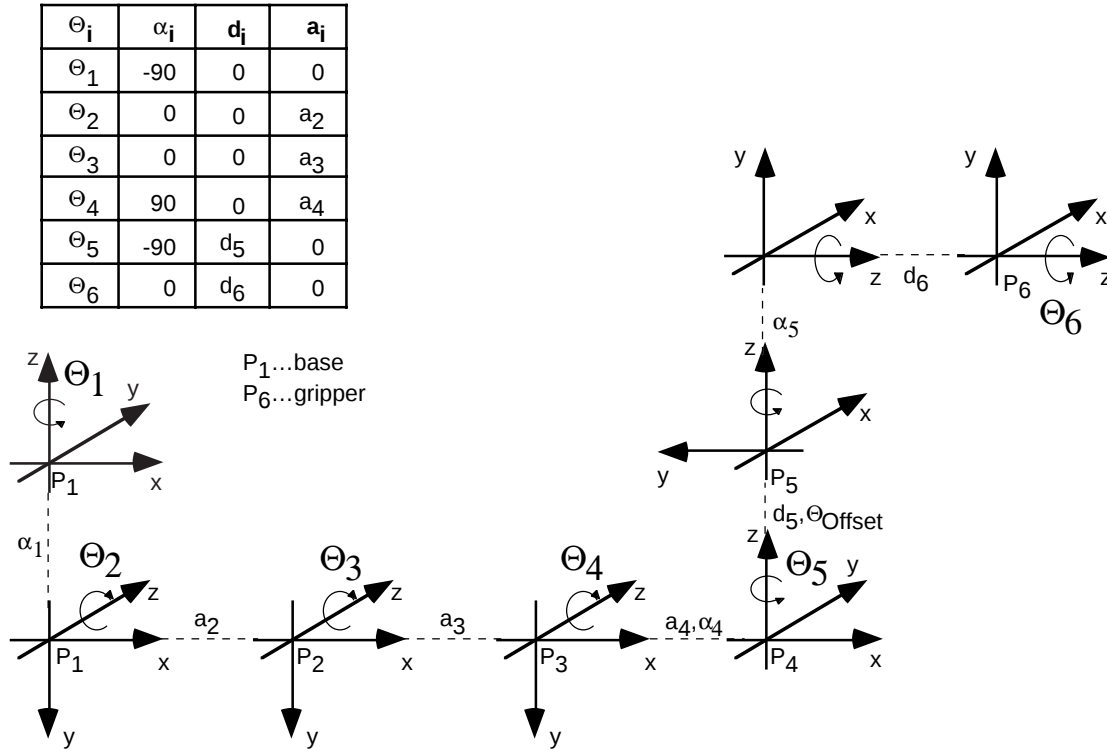
$P \dots 1 \times 3$ – matriz de perspectiva

$S \dots 1 \times 1$ – matriz de escala

Para más informaciones se remite a la fuente bibliográfica [4].

5.4.1.1. Transformación directa

El modelo de la cadena cinemática se efectúa con la determinación de varios sistemas de referencia en las articulaciones según las convenciones de Denavit-Hartenberg.

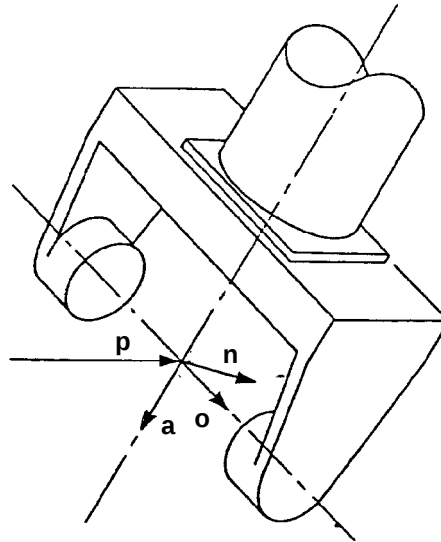


Cadena cinemática del robot GRIPS según la convención DH

Después de la definición de los parámetros y la multiplicación de las matrices resulta una matriz 4x4 (*frame de la pinza*), que contiene la posición y la orientación de la pinza en coordenadas del mundo.

$$E = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

El vector (p_x, p_y, p_z) representa la posición y los demás parámetros la orientación, donde significa *n normal*, *o orientation* y *a approach*. Se puede entender estos nombres muy bien contemplando la siguiente figura.



Sistema de referencia de la pinza con los vectores n , o , a

Se tiene ahora un método matemático para a partir de los ángulos del robot poder deducir la posición y orientación de su pinza. Esto se llama transformación directa. Pero nos interesa más el problema de la cinemática inversa (transformación inversa), es decir, calcular a partir de las coordenadas cartesianas los ángulos de las articulaciones. Esto es en general mucho más complicado, porque la inversión de las matrices y la solución del sistema de ecuaciones es casi imposible o no practicable en un tiempo razonable (sistema de ecuaciones trascendente con 12 ecuaciones y 6 incógnitas).

5.4.1.2. Transformación inversa

Para conseguir una formulación completa de la cinemática inversa, que se necesita también para el simulador TOROS, se usa un método desarrollado en 1979 Richard P. Paul a partir de la convención Denavit-Hartenberg. Este método se basa en una multiplicación sucesiva con una determinación siguiente de ángulos o sumas de ángulos a partir de valores conocidos del frame de la pinza. Con los parámetros del frame de la pinza resultan:

$$\Theta_1 = \arctan\left(\frac{p_y}{p_x}\right)$$

$$(\Theta_2 + \Theta_3 + \Theta_4) = \arctan\left(\frac{a_z}{C_1 a_x + S_1 a_y}\right)$$

$$\Theta_3 = \arccos\left(\frac{(C_1p_x + S_1p_y - a_4C_{234} - d_5S_{234})^2 + (-p_z - a_4S_{234} + d_5C_{234})^2 - a_2^2 - a_3^2}{2a_2a_3}\right)$$

$$\Theta_{23} = -\arctan\left(\frac{e}{f}\right) + \arctan\left(\frac{d}{-\sqrt{e^2 + f^2 - d^2}}\right)$$

e, f, d ... ver el cálculo detallado en el apéndice

$$\Theta_4 = \Theta_{234} - \Theta_{23}$$

$$\Theta_2 = \Theta_{23} - \Theta_3$$

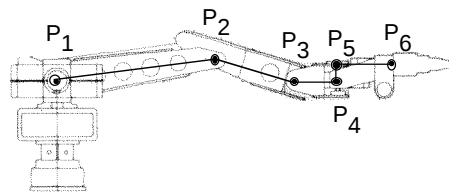
$$\Theta_6 = \arctan\left(\frac{C_4(C_{23}[C_1o_xS_1o_y] + S_{23}[-o_z]) + S_4(-S_{23}[C_1o_xS_1a_y] + C_{23}[-o_z])}{C_4(C_{23}[C_1n_xS_1n_y] + S_{23}[-n_z]) + S_4(-S_{23}[C_1n_xS_1n_y] + C_{23}[-n_z])}\right)$$

El método calculado detalladamente se encuentra en el apéndice B.1. Un problema especialmente difícil es evitar dividir por un seno o un coseno, porque con ángulos donde estas funciones trigonométricas son cero o cerca cero, resultan inexactitudes enormes. Además se tiene que tener en cuenta que las funciones trigonométricas valen únicamente en un intervalo específico.

5.4.1.3. Prueba de las ecuaciones en el simulador TOROS

Para verificar estos cálculos complejos se ha utilizado el programa matemático MAPLE. Por otra parte el modelo físico y cinemático se ha comprobado en el simulador para robots TOROS. Este simulador necesita las ecuaciones cinemáticas inversas en una forma completa pero calcula la transformación directa por si mismo vía multiplicaciones de las matrices DH. Así se pueden verificar tanto las ecuaciones de la transformación directa leyendo los parámetros del frame de la pinza como las de la inversa introduciendo una trayectoria en coordenadas cartesianas. Sin embargo TOROS supone siempre el robot modelado mediante la convención de Denavit-Hartenberg.

Para dar una idea de la posición se muestra los *frames* en la siguiente figura. Tanto la definición de los componentes mecánicos que se definen en forma de poliedros compuestos de polígonos que crecen en una dirección del espacio, como las ecuaciones cinemáticas inversas en lenguaje C se encuentran en el apéndice (B.2/B.3). Con estos datos, TOROS construye la representación del robot que se ve en el apéndice B.9.



Posición de los 'frames' de robot GRIPS

5.4.2. Descripción del espacio como una matriz 3D

Lo que nos interesa a continuación es obtener una formulación adecuada del espacio de trabajo, que permita decidir rápidamente si un *objectlocation* es asequible o no. La descripción está relacionada estrechamente con el problema de la cinemática inversa, porque esta sólo tiene sentido en el espacio de trabajo del robot (Becquet). Una descripción con una intergral de volumen solo es posible para cinemáticas muy sencillas o con muchas restricciones. Además no existen métodos para distinguir entre varias orientaciones. Entonces sólo queda la posibilidad de entrar los parámetros de cada *objectlocation* en las ecuaciones cinemáticas inversas y verificar si los ángulos obtenidos están dentro de límites particulares. El problema de este método es que se necesita mucho tiempo para calcular las ecuaciones trigonométricas complejas.

El algoritmo busca secuencialmente colocaciones posibles, variando el posicionamiento del sistema en cada posición discreta del intervalo de búsqueda. Para 30 objetos y un intervalo con las dimensiones $v_x = v_y = v_z = 20$ se tendría que calcular las ecuaciones inversas ≈ 240.000 veces. Uno puede imaginarse que resultan necesidades de tiempo no tolerables. Para evitarlo se usa el siguiente método:

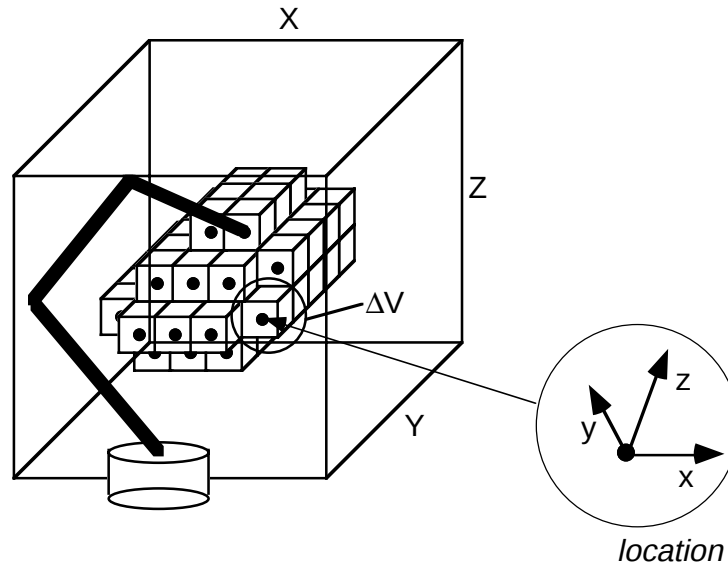
- Discretización del espacio de trabajo en unidades de volumen DV, que son representadas por un *location* en su medio. Si este *location* es asequible, todo punto en DV con la misma orientación es asequible
- Cálculo apriori de espacios de trabajo considerando distintas orientaciones con una almacenamiento óptimo

Las distancias entre los puntos medios de las unidades de volumen en las direcciones x, y y z son iguales. Esto se puede expresar mediante la resolución res:

$$\text{res}_x = \text{res}_y = \text{res}_z = \text{res}$$

Con las dimensiones X,Y y Z del espacio de trabajo se obtienen n_V unidades de volumen.

$$n_V = \frac{X}{\text{res}_x} \cdot \frac{Y}{\text{res}_y} \cdot \frac{Z}{\text{res}_z} = \text{res}^{-3} \cdot XYZ$$



Discretización del espacio de trabajo en unidades de volumen

Si se tiene en cuenta la orientación para la descripción del espacio de trabajo, resultan en general cuerpos que no son simétricos en rotación y presentan superficies cóncavas. Una partición del espacio de trabajo en dos planos bidimensionales no es posible, porque resultarían inexactitudes enormes. Entonces nos queda solamente una descripción orientada al volumen.

Esta manera del almacenamiento tiene naturalmente la desventaja de que las necesidades de memoria crecen rápidamente con una dependencia directa de la resolución res . Se tiene entonces que se obtiene mayor velocidad y menores necesidades de memoria siempre a costa de una discretización grande y una exactitud baja. A pesar de todo se muestra que con una técnica de almacenamiento inteligente se necesita solo unos kbytes por espacio de trabajo. Con un equipamiento de memoria de los computadores de unos Mbytes (tendencia creciente) esta no parece una restricción muy grave.

Este almacenamiento en una matriz 3D puede ser usado para una visualización del espacio de trabajo.

En nuestro caso especial de una división en orientaciones discretas todos los parámetros del frame de la pinza, excepto los de posición, están determinados. El cálculo se basa en una asignación sucesiva de las posiciones de todos los n_v *locations* distintos que representan las unidades de volumen. Ahora todos los parámetros del frame de la pinza son conocidos y se puede verificar la asequibilidad. Para el almacenamiento, valores de 1(asequible) y 0 (no asequible) son suficientes. La figura siguiente muestra un corte de un espacio de trabajo discreto.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1	0	0	0	0	0
0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	1	1	1	1	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0	0	0	1	1	1	1	0
1	1	1	1	1	1	1	0	0	0	0	1	1	1	1
1	1	1	1	1	1	1	0	0	0	0	1	1	1	1
1	1	1	1	1	1	1	0	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0	0	0	0	1	1	1
1	1	1	1	1	1	0	0	0	0	0	0	0	1	0
0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

corte de un espacio discretizado

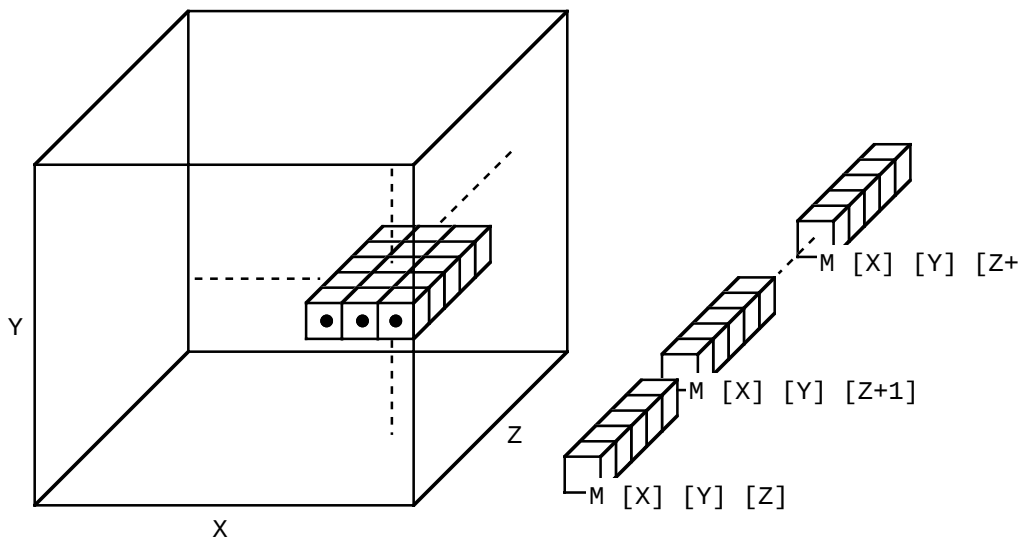
Se consigue como resultado una descripción del espacio de trabajo para cada una de las orientaciones. Esta formulación se llama descripción mediante VOXEL (Volume Pixel), que amplía el PIXEL de la descripción bidimensional a la tercera dimensión. En el apéndice B.8 fueron visualizados algunos espacios de trabajo con

VOXEL (AutoCAD). El cálculo se ha realizado con el programa MAPLE, cuyo código se encuentra también en el apéndice (B.7).

5.4.3. Técnicas de almacenamiento

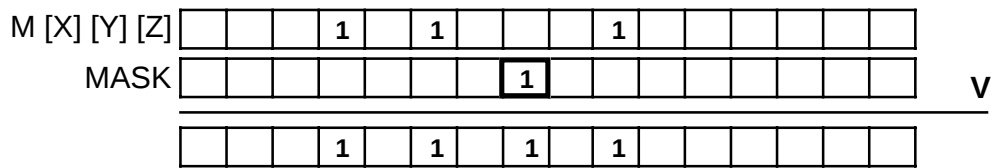
5.4.3.1. Almacenar aprovechando cada bit

Debido a que el espacio de trabajo es descrito por los valores discretos 0 y 1, no parece muy oportuno, almacenar 0 y 1 con números de dos o cuatro bytes. Aquí es lógico aprovechar mejor la memoria escribiendo bit por bit para reducir considerablemente las necesidades. En lo que sigue se supone que las variables X, Y y Z tienen cuatro bytes (32 bit). Se define una matriz tridimensional M de las dimensiones X, Y y Z que depende de la división en unidades de volumen DV. Ahora se puede usar cada uno de los 32 bit en una dimensión de la matriz definiendo una máscara adecuada y escribiendo o leyendo el bit en el sitio correspondiente. La siguiente figura muestra el uso de los bit individuales en la dimensión Z.



Almacenamiento aprovechando cada bit

La máscara está llena de ceros excepto en el sitio donde se quiera leer o escribir un bit. Así se puede establecer el valor 1 (SET) o 0 (RESET) con operaciones booleanas sencillas. La operación SET es aclarada en la figura siguiente con la combinación OR (campos vacíos corresponden a ceros lógicos)



Un ejemplo para la realización de SET, RESET y TEST se muestra en el siguiente código en C

```

/*****
short int x,y ;
long int z ;
int a,b ;
boolean VOXEL ;

a = z / 32;
b = z % 32;

Mask = 1 << b ;
W = M [x][y][a] ;

/** SET */
W |= Mask ;

/** Reset */
W &= Mask ;

/** TEST if VOXEL is true*/

if ((W & Mask) != 0)
    VOXEL = true;

*****/
```

Otra posibilidad para realizar esta estructura en C es la definición de campos de bits. La ventaja es el acceso sencillo a los bits correspondientes.

```
/******
```

```
struct BIT {  
  
    unsigned bit_1 : 1;  
    unsigned bit_2 : 1;  
    unsigned bit_3 : 1;  
    ...  
    unsigned bit_32 : 1;  
  
};
```

```
struct BIT M[X][Y][Z]
```

```
/*** SET Bit_n ***/
```

```
M[x][y][a].bit_n = 1 ;
```

```
/*** RESET Bit_n ***/
```

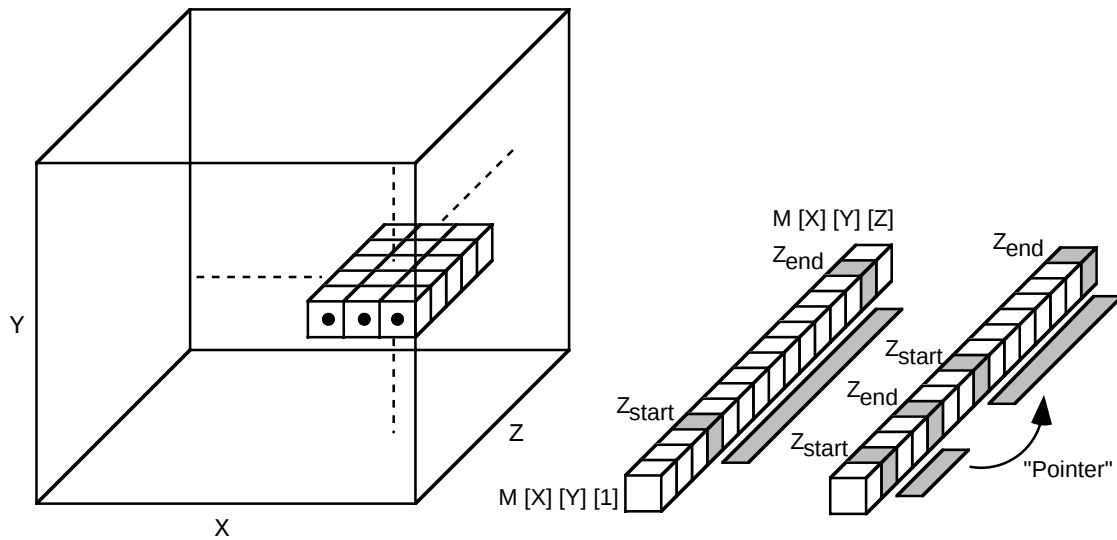
```
M[x][y][a].bit_n = 0 ;
```

```
/******
```

Con una resolución de 5cm se necesita para almacenar un espacio de trabajo del robot GRIPS una media de 5kbyte de memoria.

5.4.3.2. Almacenar en una estructura de punteros

El espacio de trabajo será en razón de la cinemática de los robots un espacio continuo . Esto significa que las unidades de volumen nunca aparecen solas y que tienen asequible, por lo menos en una dimensión del espacio, unidades vecinas de volumen. Se puede aprovechar esta aparición frecuente almacenando sólo el principio y el fin de una fila de unos y suponer que dentro de este intervalo todas las unidades de volumen son asequibles. Para este almacenamiento es especialmente adecuado una matriz de una estructura de punteros. Aquí cada espacio de trabajo con $res = 5cm$ ocupa aproximadamente 20kbyte.



Almacenamiento con una estructura de punteros

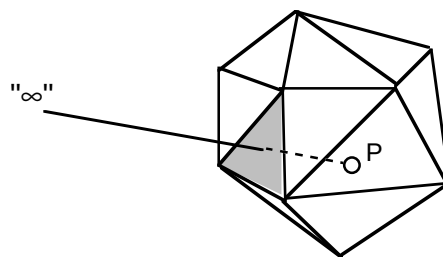
Un ejemplo de la realización de esta estructura en C se muestra el siguiente código:

```
/******  
boolean first_element  
typedef struct pnode {  
    int Z_start, Z_end ;  
    struct pnode *next ;  
} PNODE ;  
first_element = 1;  
PNODE *M[X][Y] ; *aux ; * aux_old  
  
for (x=x_min; x<x_max;x++) {  
    for (y=y_min; y<y_max;y++) {  
        aux_old = malloc(sizeof (PNODE) )  
        for (z=z_min; z<z_max;z++) {  
            if (start_reachable(x,y,z) == 1)  
                { aux=malloc( sizeof (PNODE))  
                  aux -> Z_start = z  
                  if (first_element)  
                      { first_element = 0;  
                      }  
                  aux_old = M[x][y] -> next = aux ;  
                }  
        }  
    }  
}
```

```
else
    aux_old -> next = aux }
if (end_reachable(x,y,z) == 1)
    aux -> Z_end = z ;
}
}
}

/*****/
```

Otra posibilidad imaginable sería el almacenamiento de la envolvente. Esta forma de describir el espacio de trabajo no está orientada al volumen, sino a la superficie. Hay que encontrar un método para mostrar, si un punto se encuentra dentro o fuera de la envolvente. La figura siguiente presenta un bosquejo de una envolvente descrita por triángulos (*triangularización* - el método usado más frecuentemente)



Envolvente descrita por triángulos

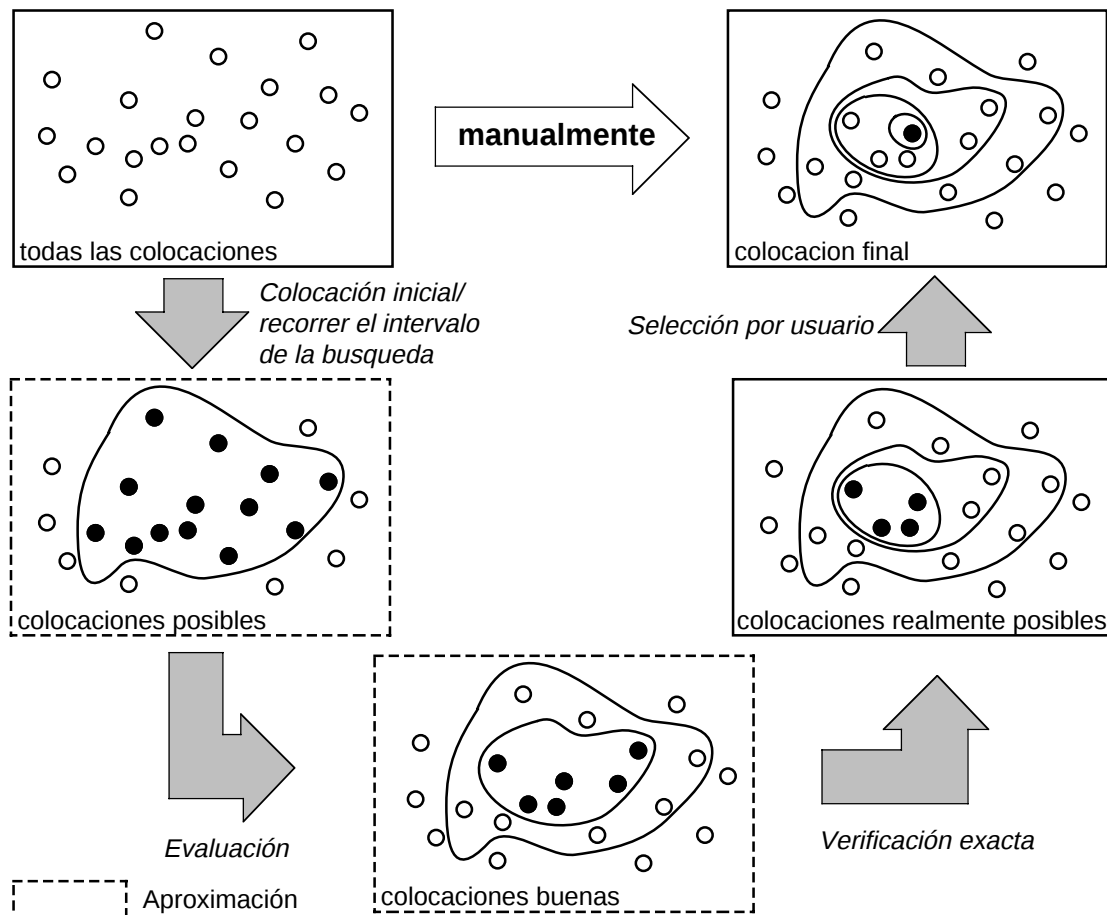
Una posibilidad para la verificación de si un punto se encuentra fuera o dentro del cuerpo sería determinar, cuántos elementos de superficie corta una línea de un punto muy lejano al punto de prueba. Con un numero de cortes par el punto está fuera y con un número impar está dentro del cuerpo. Algunas de las razones por las que este método no nos sirve aquí, son:

- Los métodos exactos para el cálculo de la envolvente son muy complejos e intensivos en tiempo (*contour tracing* fi *boundary detection* fi *triangulation*)
- Probar si un punto está dentro o fuera del cuerpo cuesta mucho tiempo, ya que se tiene que verificar si cada elemento de superficie ha sido cortado.
- No hay ningún ahorro de memoria comparado con la solución con punteros

5.5. Algoritmos de colocación

5.5.1. Estructura

Se exige del algoritmo que proporcione al usuario algunas colocaciones buenas para el sistema frente al poste. Una colocación manual realizada directamente por el hombre basándose en experiencias y algunos ensayos, en la mayoría de los casos no es óptima. Si se encuentra una colocación posible es muy difícil decir si hay más e incluso mejores soluciones. Estas desventajas se evitan con una estructura jerárquica y secuencial del problema. La determinación de una colocación final se muestra esquemáticamente en la figura siguiente:



Con una variación sucesiva de la posición del sistema en un intervalo determinado se busca primero todas las colocaciones posibles. A partir de esto, se reduce el número de las soluciones progresivamente hasta quedar sólo un número manejable de colocaciones.

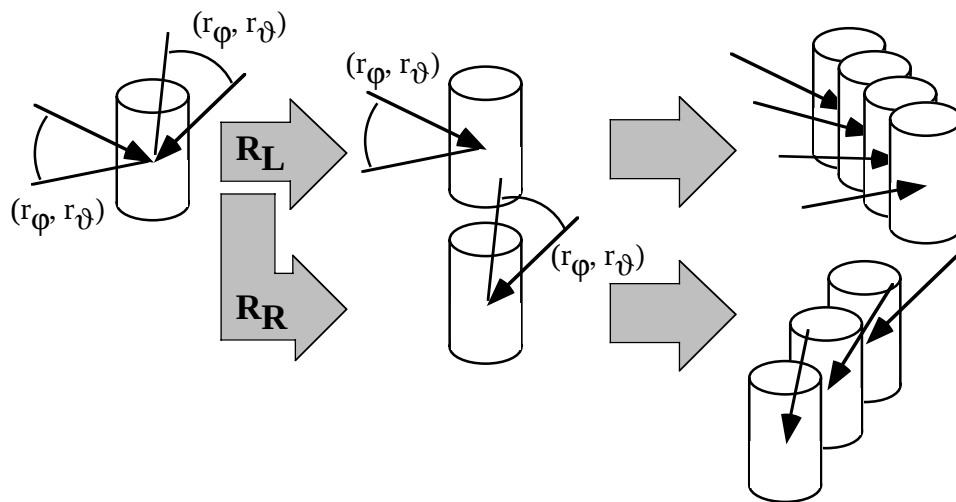
5.5.2. Preparación de los datos

5.5.2.1. División en objetos simples

En realidad el algoritmo de colocación sólo necesita los datos de objetos únicos con una posición, una orientación, una especificación del robot para la manipulación, el peso y tipo del objeto. Es decir, que en el primer paso tenemos que cambiar el formato anterior de los datos a un nuevo formato, basado en parámetros de objetos únicos. Son necesarias las siguientes descomposiciones :

- División de un ZAO en dos ZAOs con una especificación fija para cada robot respectivamente.
- Hay que resolver la rotación o la traslación en un número de objetos con posición y orientación discreta

Como ya hemos dicho al principio no queremos profundizar más en este problema. En el bosquejo siguiente se muestra esquemáticamente la división de un ZAO en dos ZAOs y luego la resolución de la rotación en un número discreto de objetos. Ya que en este ejemplo no se considera un vector de una traslación, todos los objetos tienen la misma posición.

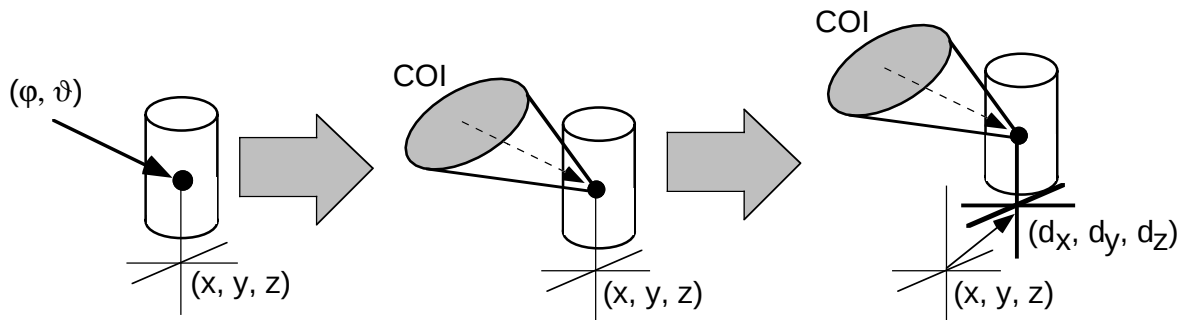


División de un ZAO en objetos únicos con una orientación discreta

5.5.2.2. Clasificación de los objetos en los intervalos de orientación y en la trama de la resolución

Debido a que se almacena el espacio de trabajo de forma discreta con una resolución res , los objetos tienen que ser también clasificados en esta trama de distancias. Por eso se divide el espacio total del mundo en unidades de volumen con resolución res y a cada *objectlocation* se añade el parámetro adicional de la posición (d_x, d_y, d_z) del punto medio que representa la unidad de volumen. De esta forma se facilita mucho la verificación de cual de las unidades de volumen del espacio de trabajo contiene al objeto. Hacer la discretización del objeto un cada paso de la búsqueda secuencial aumentaría el tiempo de ejecución del algoritmo considerablemente.

Para reducir la cantidad de los datos almacenados, hemos hecho simplificaciones en la descripción del espacio de trabajo, realizando una división de las orientaciones en intervalos particulares. Entonces, es necesario clasificar cada objeto respectivamente en su intervalo. Esta información se codifica (*COI*) y se añade como parámetro del objeto.



Clasificación del objeto en intervalos de orientación y trama de la resolución

Con la preparación de los datos resulta un numero de objetos descritos con los parámetros siguientes:

Posición exacta			Posición discreta			Orient. exacta		g	COT	COI	Robot	
x	y	z	d _x	d _y	d _z	j	J				R _L	R _R
-	-	-	-	-	-	-	-	-	EAO	-	1	1
-	-	-	-	-	-	-	-	-	EAO	-	0	1
-	-	-	-	-	-	-	-	-	EAO	-	1	0
-	-	-	-	-	-	-	-	-	ZAO	-	0	1
-	-	-	-	-	-	-	-	-	ZAO	-	1	0

5.5.2.3. *Cargar los datos en la memoria principal*

Ya que la colocación está asistida por computador no debe poder ser solamente utilizada en el laboratorio con ordenadores potentes, sino también in situ, y esto hay que considerarlo para el desarrollo de los algoritmos. Debido a que el espacio de trabajo es dado en forma de VOXEL, se ofrece la posibilidad de operaciones rápidas de memoria. Solo se verifica si un campo de bit está a 0 o a 1. Con un equipamiento suficiente de memoria principal, se puede cargar de una vez todos los espacios de trabajo que se necesitan para la colocación. Así se evita una carga lenta durante la búsqueda de la colocación.

5.5.3. *Primera prueba de plausibilidad*

Antes de empezar con el algoritmo de colocación, se verifica primero si la introducción de los objetos por el usuario puede conducir a una solución. Esto permite ahorrar tiempo, porque entradas de datos sin sentido son reconocidas inmediatamente y no son aceptadas por el programa. Para la verificación se consideran sólo las posiciones de los objetos separados en EAOs y ZAOs y se definen dos criterios necesarios para una colocación posible. En la formulación del primer criterio se hacen simplificaciones enormes, mientras que el segundo criterio se basa en una consideración bastante exacta (perfeccionamiento sucesivo de las condiciones).

Claro que no se puede deducir la existencia de una colocación posible porque una prueba haya tenido éxito, ya que no se han tenido en cuenta las orientaciones. Sin embargo antes de la entrada en el algoritmo ya se han eliminado entradas que no pueden llevar a una solución.

5.5.3.1. Primer criterio necesario

Los objetos forman un espacio que está caracterizado por sus puntos extremos en las direcciones x, y, z. Estos puntos para la dirección x se llaman K_{min}, K_{max} , para la dirección y L_{min}, L_{max} y para la dirección z M_{min}, M_{max} . De igual forma para los ZAOs se les denominan $K_{coordmin}, K_{coordmax}, L_{coordmin}, L_{coordmax}, M_{coordmin}$ y $M_{coordmax}$. [v.fig.]

Para la primera prueba, si los puntos son asequibles por los robots, se hace la siguiente suposición: Las posiciones de los objetos al igual que el espacio de trabajo utilizable idealizado, son circunscritos con paralelepípedos. A partir de esto se puede deducir para la asequibilidad de los objetos el siguiente criterio:

- Las dimensiones del paralelepípedo ($\Delta b, \Delta h, \Delta t$) que circunscribe los objetos tienen que ser mas pequeñas que las del espacio de trabajo utilizable idealizado

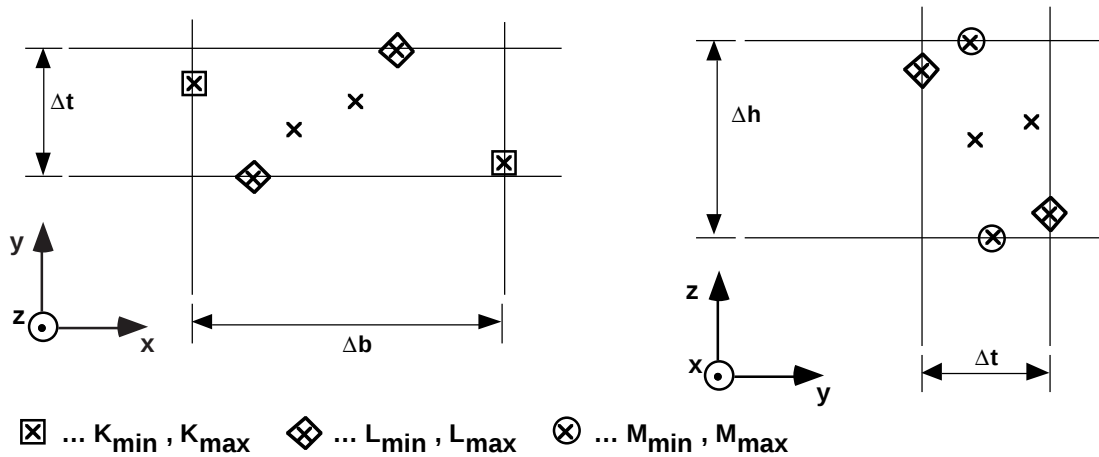
$$1. (\Delta b \leq b_{IAR}) \& (\Delta t \leq t_{IAR}) \& (\Delta h \leq h_{IAR})$$

$$2. (\Delta b_{coord} \leq (2r - d_R)) \& (\Delta t_{coord} \leq t_{IAR}) \& (\Delta h_{coord} \leq h_{IAR})$$

A partir de los puntos extremos es muy fácil calcular la anchura, la profundidad y la altura.

$$\begin{array}{ll} \Delta b = K_{max} - K_{min} & \text{bzw.} \quad \Delta b_{coord} = (K_{coordmax} - K_{coordmin}) \\ \Delta t = L_{max} - L_{min} & \Delta t_{coord} = (L_{coordmax} - L_{coordmin}) \\ \Delta h = M_{max} - M_{min} & \Delta h_{coord} = (M_{coordmax} - M_{coordmin}) \end{array}$$

Los puntos extremos, anchura, profundidad y altura son aclarados en la figura siguiente que presenta dos cortes del espacio contemplado.



En el caso de que exista sólo uno o ningún objeto para manipulaciones coordinadas (ZAOs), las variables Db_{coord} , Dh_{coord} , Dt_{coord} serán cero o no estarán definidas y no podrán ser utilizadas en el algoritmo. Por eso hay que contemplar estos casos separadamente allí donde el algoritmo se base en ellas.

5.5.3.2. Segundo criterio

La primera simplificación, con la descripción del espacio de trabajo en la forma de un paralelepípedo, es muy impreciso, porque en verdad el espacio de trabajo es simétrico en rotación. Por eso se usa aquí la descripción que dada por el fabricante en dos vistas ortogonales. En razón de la simetría de rotación hay que hacer una transformación en coordenadas cilíndricas. El criterio resulta entonces:

- Todos los objetos tienen que encontrarse en la vista del plano xy y en la vista rh

Se puede entonces serializar la verificación, es decir que se busca primero una colocación de todos los objetos proyectados en el plano xy (variación en la dirección x e y) y luego en el plano rh (variación en z). Claro que también se tiene que verificar, si están también en el espacio de trabajo utilizable.

En la siguiente figura se explica superficialmente el algoritmo. La descripción dada en dos planos es formulada con ecuaciones de círculos y líneas. Estas definiciones de círculos y líneas se encuentran en el apéndice B.4.

Para el posicionamiento inicial se distingue entre tres casos:

- $\Delta b_{\text{coord}} \neq 0$... varios ZAOs
- $\Delta b_{\text{coord}} = 0$... un objeto ZAO
- Δb_{coord} no definida ... ningún objeto ZAO

- Ninguna restricción del espacio de trabajo total

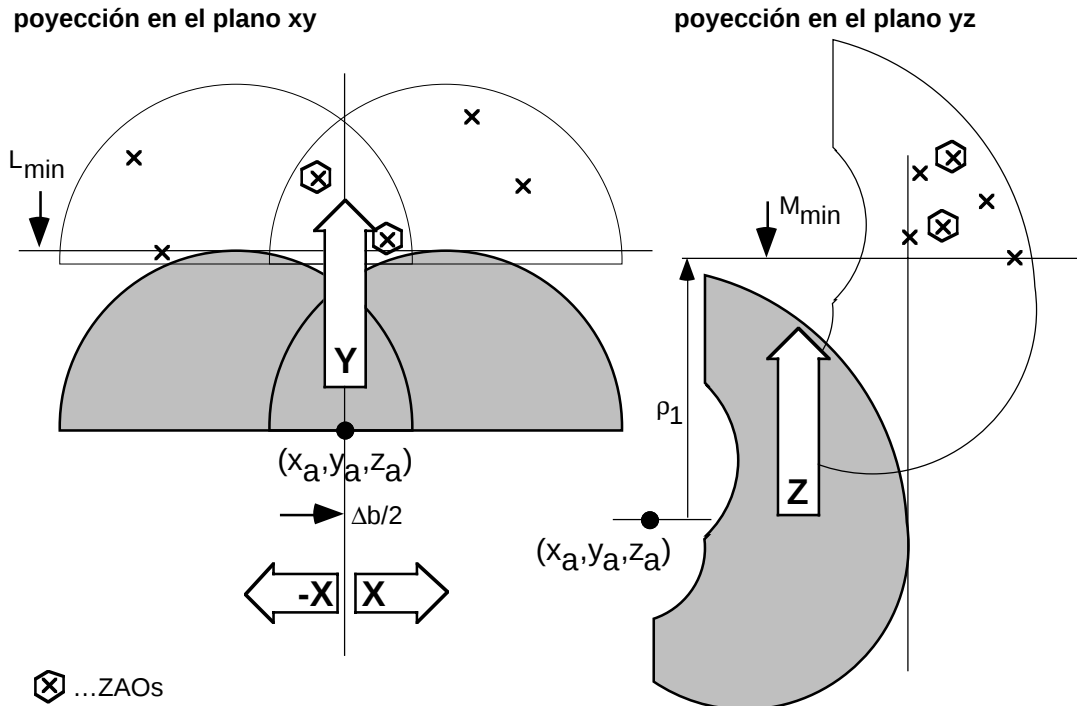
$\Delta b_{\text{coord}} \neq 0$ [v.fig.]	$x_a = x_{K_coord \max} - \Delta b_{\text{coord}} / 2$ $y_a = y_{L \min} - \rho_1$ $z_a = z_{M \min} - \rho_1$
$\Delta b_{\text{coord}} = 0$	$x_a = x_{\text{coordObj}}$ $y_a = y_{L \min} - \rho_1$ $z_a = z_{M \min} - \rho_1$
Δb_{coord} no definida	$x_a = x_{K \max} - \Delta b / 2$ $y_a = y_{L \min} - \rho_1$ $z_a = z_{M \min} - \rho_1$

- Restricción del espacio de trabajo total

$\Delta b_{\text{coord}} \neq 0$	$x_a = x_{K\text{coord} \max} - \Delta b_{\text{coord}} / 2$ $y_a = y_{L \min} - (y_{\text{PIAR}} + t_{\text{IAR}})$ $z_a = z_{M \min} - (z_{\text{PIAR}} + h_{\text{IAR}})$
$\Delta b_{\text{coord}} = 0$	$x_a = x_{\text{coordObj}}$ $y_a = y_{L \min} - (y_{\text{PIAR}} + t_{\text{IAR}})$ $z_a = z_{M \min} - (z_{\text{PIAR}} + h_{\text{IAR}})$
Δb_{coord} no definida	$x_a = (x_{K \max} - \Delta b_{\text{coord}} / 2) + (x_{\text{PIAR}} - b_{\text{IAR}} / 2)$ $y_a = y_{L \min} - (y_{\text{PIAR}} + t_{\text{IAR}})$ $z_a = z_{M \min} - (z_{\text{PIAR}} + h_{\text{IAR}})$

En el caso de la existencia de más que un objeto ZAO la coordenada x_a se define en el punto medio de los puntos extremos, porque el espacio de trabajo común es simétrico y resulta entonces una buena oportunidad para colocar los objetos dentro del espacio de trabajo común. Solamente si falla la colocación con esta x_a fija se varía x por Dx y $-Dx$ y se comienza el algoritmo con esta nueva x .

Para los demás casos atípicos (solo un ZAO/ningún ZAO) se realiza un cálculo modificado, lo cual se explica en el apéndice B.5.

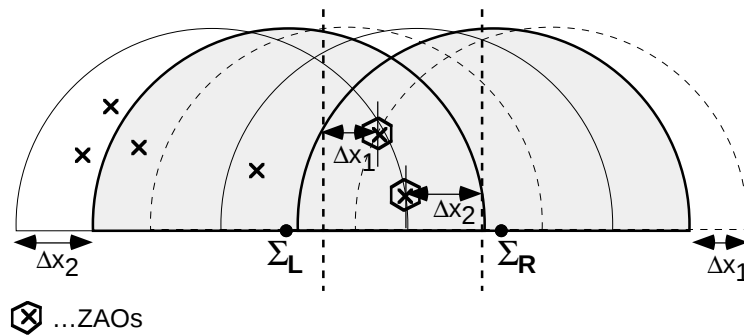


Colocación inicial para varios ZAOs

Se espera de una determinación inteligente de la colocación inicial, que en la mayoría de los casos se tenga que variar sólo en y y z para cumplir el criterio. En este caso no es decisivo encontrar todas las colocaciones posibles, sino que se puede interrumpir el algoritmo después de encontrar una.

Pero si se tiene que variar x, se puede reducir también el tiempo de la búsqueda con una determinación inteligente de Dx.

Un ejemplo cualitativo se muestra en la figura siguiente:

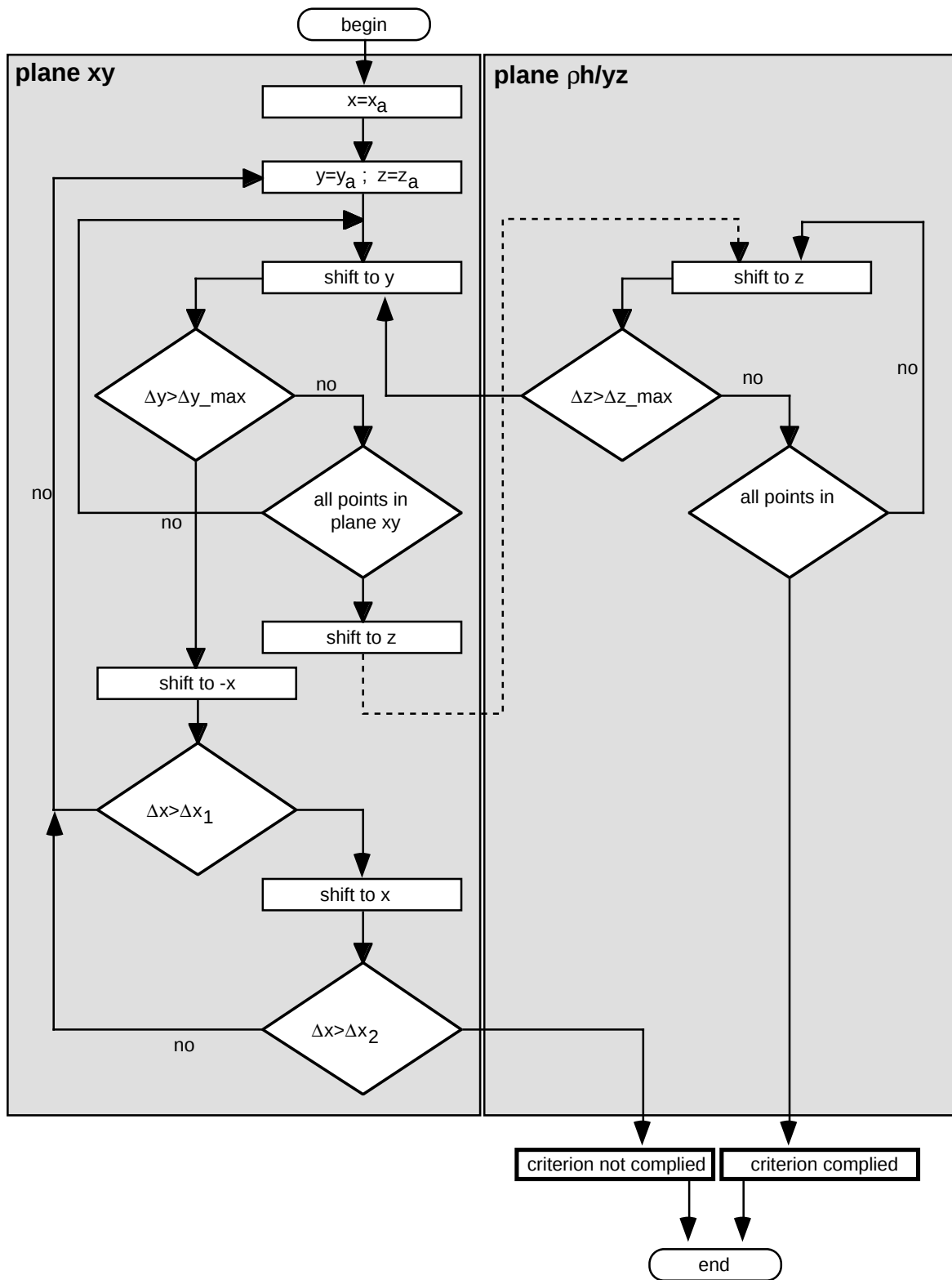


determinación de Dx

El algoritmo se presenta en forma de un flujograma, cuya nomenclatura es la siguiente:

x_a, y_a, z_a	... Colocación inicial del sistema. Varía según el caso
Dy_{max}, Dz_{max}	... Intervalo máximo en las dimensiones y, z. Para la dimensión x se hace una contemplación especial
Dx_1, Dx_2	... Intervalo de variación en x (Dx_2) y -x (Dx_1) calculado anteriormente

Se ve muy bien la partición del problema tridimensional en dos bidimensionales. Sólo se alcanza la verificación de si un punto está dentro o fuera del área del plano yz/rh, tras la realización con éxito de una prueba en el plano xy. Esto lleva tal vez a un aumento de la velocidad para la verificación, porque si no se encuentra una colocación posible en xy, se puede suprimir la variación en la tercera dimensión.



La pregunta de si un punto se encuentra en el plano xy, se formula ahora de forma analítica fácilmente .

Para la nomenclatura de los sistemas de referencia auxiliares y las componentes relativas x, y, z se presentan las siguientes notas.

Coordenadas del punto P relativas a {S}:

$$\begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix}_{\{\Sigma\}} = \begin{pmatrix} x_{p,\{\Sigma\}} \\ y_{p,\{\Sigma\}} \\ z_{p,\{\Sigma\}} \end{pmatrix} = \vec{p} - \vec{\Sigma}$$

$\vec{p}$...vector de posición

$\vec{\Sigma}$...vector referido al sistema Σ

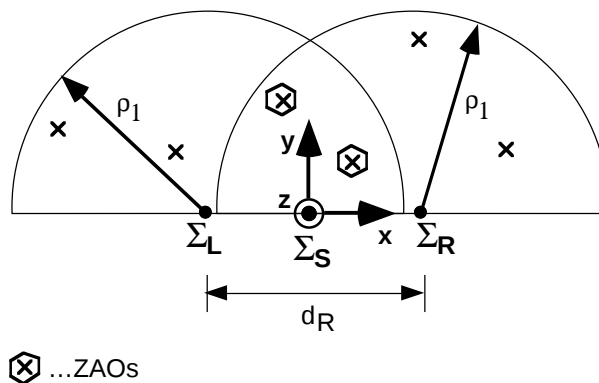
La definición de los sistemas de referencia auxiliares se realiza a través de una traslación relativa al sistema de referencia S (sin un cambio de orientación)

$$\vec{\Omega} = \vec{\Sigma} + \vec{t}$$

$\vec{t}$...vector de traslación

$\vec{\Omega}$...vector referido al sistema auxiliar Ω

A partir de la definición del plano xy como dos semicírculos, se puede fácilmente verificar si un punto (la proyección del punto en el plano xy) está dentro o no. Con la ayuda de las ecuaciones del círculo se define una desigualdad que tiene que cumplir el punto O.



Representación del plano xy con objetos y espacio de trabajo del sistema

El objeto para manipulaciones con un brazo (EAO) tiene que estar en el semicírculo 1 o en el semicírculo 2

$$\left(x_{O,\{\Sigma_L\}}^2 + y_{O,\{\Sigma_L\}}^2 \leq \rho_1^2\right) \vee \left(x_{O,\{\Sigma_R\}}^2 + y_{O,\{\Sigma_R\}}^2 \leq \rho_1^2\right)$$

El objeto para manipulaciones coordinadas (ZAO) tiene que estar en el semicírculo 1 y en el semicírculo 2

$$\left(x_{O,\{\Sigma_L\}}^2 + y_{O,\{\Sigma_L\}}^2 \leq \rho_1^2\right) \& \left(x_{O,\{\Sigma_R\}}^2 + y_{O,\{\Sigma_R\}}^2 \leq \rho_1^2\right)$$

El mismo principio que en el plano xy se usa en el plano rh. Desgraciadamente la geometría aquí no es tan fácil y hay que distinguir si el punto esta encima o debajo del plano formado por los ejes x e y del sistema de referencia auxiliar S_H . Una vez realizada la decisión resulta un área limitada por dos arcos de círculo y una línea. Con una transformación en coordenadas cilíndricas

$$r_{\{\Sigma\}} = \sqrt{x_{O,\{\Sigma\}}^2 + y_{O,\{\Sigma\}}^2} \quad y \quad h = z_{O,\{\Sigma\}}$$

se puede formular de nuevo unas desigualdades

punto debajo del plano con $F(x,y,0)$

$$\left(r_{\{\Sigma_{L/R}\}} < \varepsilon\right) \& \left(h_{\{\Sigma_{L/R}\}}^2 + r_{\{\Sigma_{L/R}\}}^2 \geq \rho_3^2\right) \& \left(h_{\{\Omega_2\}}^2 + r_{\{\Omega_2\}}^2 \leq \rho_2^2\right)$$

punto encima del plano con $F(x,y,0)$

$$\left(r_{\{\Sigma_{L/R}\}} < \rho_3\right) \& \left(h_{\{\Sigma_{L/R}\}}^2 + r_{\{\Sigma_{L/R}\}}^2 \leq \rho_1^2\right) \& \left(h_{\{\Omega_1\}}^2 + r_{\{\Omega_1\}}^2 \geq \rho_2^2\right)$$

Claro que, como antes, se tiene que hacer esta verificación separadamente para EAOs y ZAOs con las combinaciones booleanas correspondientes.

Para la determinación de si los puntos se encuentran también en el espacio de trabajo utilizable se usa el método anterior, con una división en los planos xy e yz. Las ecuaciones son mucho más fáciles porque sólo se tienen líneas y tampoco hay que distinguir entre EAOs y ZAOs. Las desigualdades resultantes son:

1. $(x_{O,\{P_{IAR}\}} \geq 0) \ \& \ (x_{O,\{P_{IAR}\}} \leq b_{IAR})$
2. $(y_{O,\{P_{IAR}\}} \geq 0) \ \& \ (y_{O,\{P_{IAR}\}} \leq t_{IAR})$
3. $(z_{O,\{P_{IAR}\}} \geq 0) \ \& \ (z_{O,\{P_{IAR}\}} \leq h_{IAR})$

5.5.4. Colocación inicial

La elección de la situación inicial es muy importante para el algoritmo, porque la velocidad de la búsqueda de soluciones depende esencialmente de ella. Es importante entonces buscar una colocación inicial tal que se encuentre un número de soluciones tan grande como sea posible en un intervalo de búsqueda pequeño alrededor de esta primera colocación. Luego se valoran estos posicionamientos y se eligen los mejores.

5.5.4.1. Centro de gravedad de los objetos

Si se contempla el espacio de trabajo y el cuerpo circunscribiendo los objetos como cuerpos de volumen, se deben colocar los volúmenes tal que la combinación booleana AND (es decir la intersección) sea tan grande como sea posible. Además es importante que los puntos se encuentren en el centro del volumen y no muy cerca del borde del espacio de trabajo. Esto significa que las zonas centrales tienen que coincidir. Debido a que el centro de gravedad representa el centro del cuerpo, parece una buena aproximación reunir los dos centros de gravedad.

Si se contemplan los objetos como masas puntuales con el peso g , se encontrarán en zonas de mejor o de peor calidad según su peso. Estando los objetos con las masas g_{oi} en las posiciones $o_i = (x_i, y_i, z_i)$, se calcula el centro de gravedad como:

$$\bar{S}_O = \frac{\sum g_i \bar{o}_i}{\sum g_i}$$

$\bar{o}_i$...vector de posición de la masa puntual i con peso g_i

Para el cálculo del centro de gravedad del espacio de trabajo con la siguiente colocación

$$\bar{S}_{AR} = \bar{S}_O$$

hay primero que definir qué parte del espacio de trabajo y qué objetos se tienen en cuenta aquí. Se distinguen dos casos:

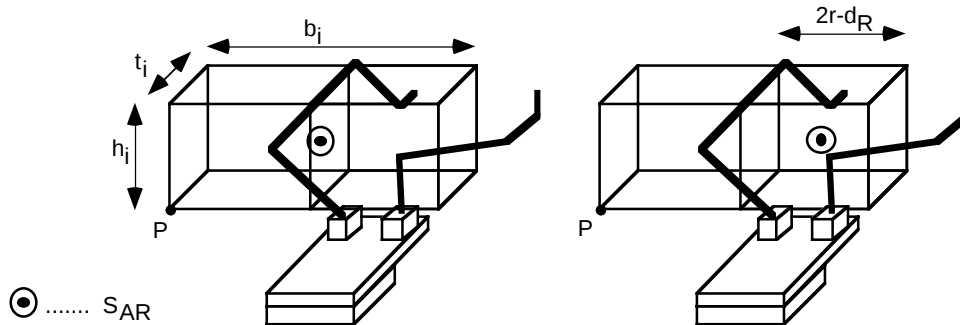
- Si existe por lo menos un objeto ZAO, se considera para la colocación inicial sólo estos objetos y el espacio de trabajo común utilizable idealizado. El centro de gravedad S_{AR} resulta

$$\bar{S}_{AR} = \begin{pmatrix} x_{S_{AR}} \\ y_{S_{AR}} \\ z_{S_{AR}} \end{pmatrix}_{\{P_{IAR}\}} = \frac{1}{2} \cdot \begin{pmatrix} 2b_{IAR} - (2r - d_R) \\ t_{IAR} \\ h_{IAR} \end{pmatrix}$$

- Si no existe ningún ZAO, se tiene en cuenta el espacio de trabajo idealizado entero y todos los objetos. Entonces el S_{AR} es:

$$\bar{S}_{AR} = \begin{pmatrix} x_{S_{AR}} \\ y_{S_{AR}} \\ z_{S_{AR}} \end{pmatrix}_{\{P_{IAR}\}} = \frac{1}{2} \cdot \begin{pmatrix} b_{IAR} \\ t_{IAR} \\ h_{IAR} \end{pmatrix}$$

La razón para esta diferenciación se basa en el tamaño mas pequeño del espacio de trabajo común y en que objetos manipulados coordinadamente exigen más variedad de orientaciones. Por eso tienen que estar en zonas centrales del espacio de trabajo.



Posición del centro de la gravedad en los casos ningún ZAO o varios ZAOs

5.5.5. Recorrer el intervalo de búsqueda

Después de encontrar una colocación inicial apropiada, hay que variar la posición del sistema en el intervalo definido. Todos los posicionamientos posibles son almacenados realizando en el próximo paso una valorización de los mismos para hacer una elección apropiada de las colocaciones finales.

5.5.5.1. Recorrer sucesivamente el intervalo de la búsqueda definido por el operario

La posibilidad más sencilla es recorrer sucesivamente el intervalo definido. Se divide el intervalo en puntos discretos, que son verificados para una colocación posible. El número de los puntos verificados es entonces:

$$n = v_x \cdot v_y \cdot v_z$$

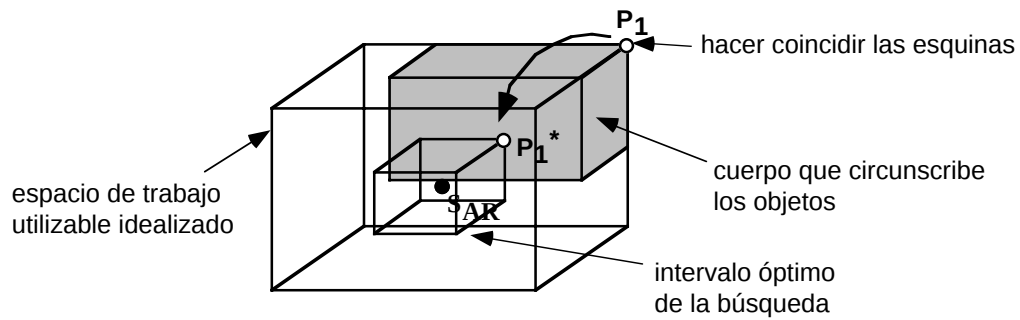
5.5.5.2. Determinación automática del intervalo óptimo de búsqueda

El problema recorrer sucesivamente es el tiempo de la búsqueda, que crece directamente cuanto más fina es la resolución y más grande el intervalo. Especialmente una resolución muy alta influye mucho en el tiempo de búsqueda.

Suponemos, por ejemplo, que un intervalo fijo es representado por un paralelepípedo rectangular con una longitud de sus aristas de 0,5 m. Entonces hay que verificar con una resolución de $d = 10$ cm, con $d = 5$ cm y con $d = 2,5$ cm para 125, 1000 y 8000 colocaciones respectivamente. Partir por la mitad la resolución significa entonces un crecimiento del tiempo de búsqueda de ocho veces.

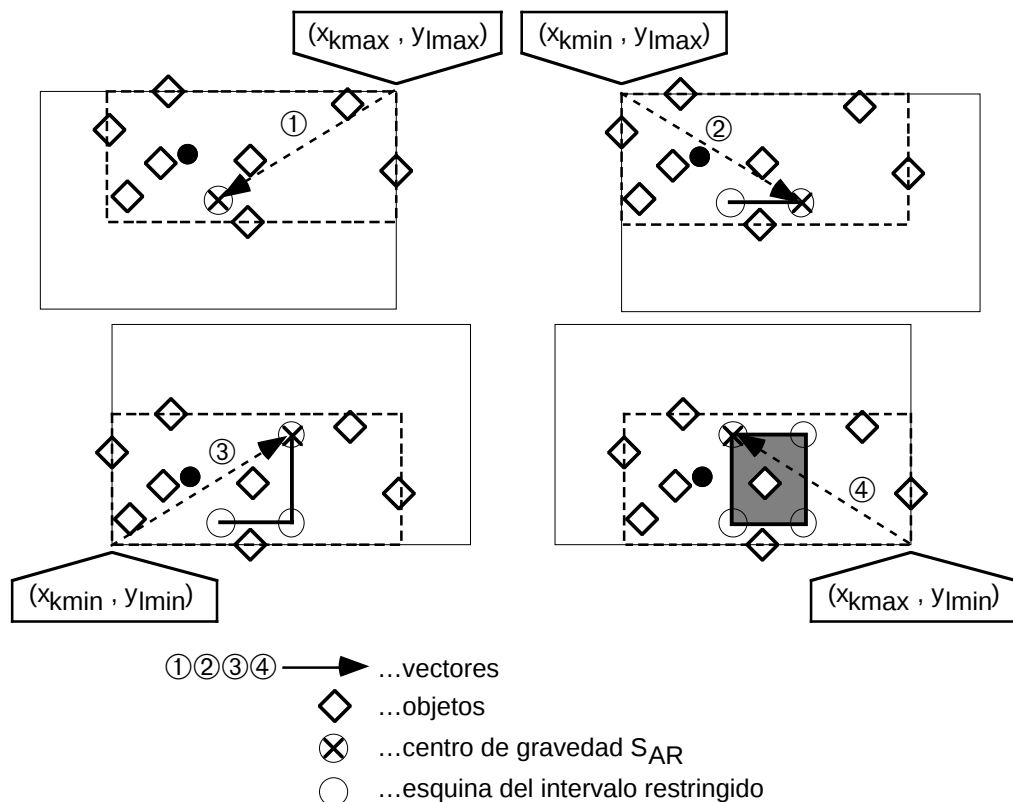
Por eso se ensaya disminuir el intervalo de la búsqueda según consideraciones geométricas sencillas, es decir que se eliminan colocaciones que nunca pueden ser posibles.

Se utiliza aquí el hecho de que los objetos tienen que encontrarse siempre en el espacio de trabajo utilizable idealizado. El intervalo resultado posee entonces respectivamente como esquinas, las posiciones del centro de gravedad del espacio de trabajo utilizable si se hacen coincidir sus esquinas con las esquinas del cuerpo que circunscribe los objetos. La figura siguiente muestra el procedimiento para una esquina (arriba a la izquierda)

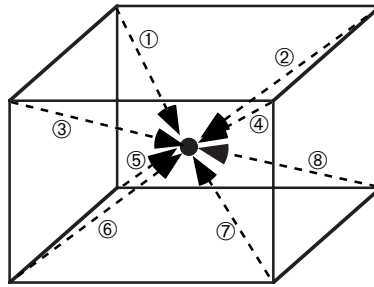


El punto P_1 determina el punto P_1^* del intervalo óptimo de la búsqueda

Ya que las coordenadas de las esquinas del cuerpo de los objetos están dadas por los valores x , y , z de K_{min} , K_{max} , L_{min} etc. la determinación del intervalo restringido es sencilla. A partir de estas esquinas se suma sólo el vector correspondiente, que va desde la esquina al centro de gravedad S_{AR} . Por claridad se muestra en la figura el método en el caso bidimensional. Se puede ampliar fácilmente al caso tridimensional.



Se obtienen, dependiendo del número de esquinas, cuatro vectores distintos y en el caso tridimensional ocho vectores.



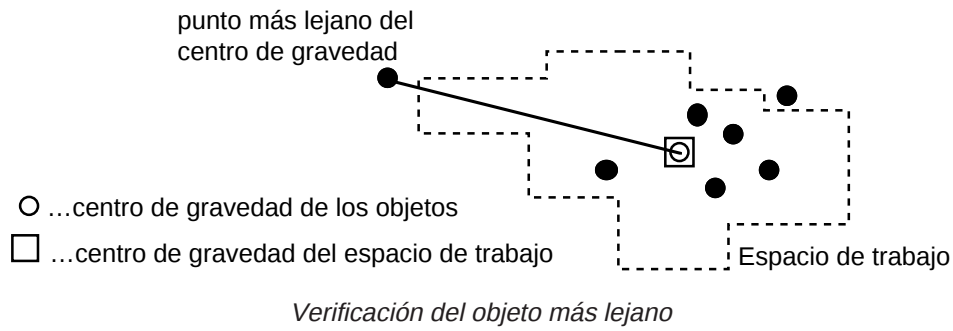
El número de las posiciones discretas en este intervalo restringido se denomina n_{aut} .

A partir del número de posiciones para verificación (n_{aut} ó n_S), se puede decidir ahora que intervalo de búsqueda se tiene en cuenta. La condición escrita matemáticamente es:

$$n = \min(n_{aut}, n_S)$$

5.5.5.3. Verificación de la asequibilidad de los objetos

Para cada una de estas n colocaciones en el intervalo de búsqueda hay que verificar el alcance de los objetos. Sin embargo en la mayoría de las pruebas de asequibilidad no es necesario contemplar todos los objetos. Una colocación falla cuando algún objeto no sea asequible y se puede cancelar entonces la verificación de los demás. Es decir, que resultarían posibilidades de disminuir el tiempo del algoritmo aprovechando este hecho. Parece, por ejemplo, una buena idea, verificar primero los objetos lejos del centro de gravedad, porque es más probable que se encuentren fuera del espacio alcanzable y por eso se puede interrumpir la verificación para esta colocación más rápidamente. El ejemplo siguiente muestra la colocación inicial (los centros de gravedad coinciden). Si se verifica el punto más lejano primero, la colocación falla después de la primera prueba para alcanzar un objeto.



5.5.6. Evaluación de la colocación

En este párrafo nos ocupamos de como se puede elegir, de entre la totalidad de colocaciones posibles, la mejor para nuestra tarea particular. No se puede definir la expresión óptima globalmente e independiente de la tarea, sino que su definición se orienta a los criterios de valorización programados. Este es entonces el sitio en el programa donde pueden ser formuladas experiencias de la realización manual mediante reglas y criterios. Se debe tener cuidado en la realización de la estructura del programa para que pueda ser modificado y ampliado fácilmente. Como resultado se consiguen entonces todas las colocaciones que se encuentran en un intervalo alrededor del máximo de la calidad. La cantidad de las soluciones depende naturalmente del tamaño de este intervalo, que puede elegir el usuario. En los siguientes apartados se distingue entre criterios dependientes e independientes del poste. La única diferencia es que en los criterios dependientes del poste hay parámetros que se cambian según el poste.

Todos los criterios se basan únicamente en la posición, es decir, sólo en las coordenadas x, y, z. La calidad de una colocación se representa por un número, que cuanto más alto mayor calidad. En principio se asigna el valor 1 a todas las colocaciones cambiándose luego dependiendo de los criterios. Resulta así una función de calidad $G_n(q_j)$ para cada una de las n colocaciones, que dependen de los parámetros q_j y de sus propios factores de peso.

$$G_n(q_j) = \prod (q_j^{v_j})$$

Los valores iniciales son:

$$p_i = 1 \quad \text{para} \quad i = 0 \dots \infty$$

Los parámetros q_j de la función de calidad son definidos también, con frecuencia, como una suma cuando un criterio tiene en cuenta más de un punto de referencia. Entonces se obtiene para el criterio j con k puntos de referencia:

$$q_j = \sum_k q_{jk}$$

En los siguientes apartados se da una idea del procedimiento de definición de los criterios mediante tres ejemplos.

5.5.6.1. Criterios independientes del poste

En razón de la distancia de seguridad y del peligro de una colisión se intenta manipular los objetos a una distancia grande. Por ejemplo, cada objeto respresenta un punto de referencia, y así se puede definir el parametro p_{1k} del primer criterio como la distancia de la base del sistema al objeto k . Si se suma para todos los objetos resulta

$$q_j = \sum_k q_{jk} = \sum_k \sqrt{(x_n - x_k)^2 + (y_n - y_k)^2 + (z_n - z_k)^2}$$

con

(x_n, y_n, z_n) ...posición de la colocación n –ésima

(x_k, y_k, z_k) ...posición del objeto k –ésimo

Otra valorización se basa en la definición de áreas preferidas. Las razones para preferir algunas zonas podrían ser:

- Exigencias ergonómicas, ya que el operario puede realizar las manipulaciones con los brazos extendidos de forma menos exacta y cómoda
- Algunas zonas del espacio pueden ser presentadas con varias vistas o con una resolución más alta por una buena posición de la cámara. En la operación con una vista directa del usuario hay zonas con una buena y una mala visibilidad causadas por la geometría del sistema o la cabina.

Se puede describir estas zonas aproximadamente por cuerpos geométricos sencillos (paralelepipedo, esfera, cilindros, conos,...) o en forma de ecuaciones tridimensionales. Cada una de estas zonas preferidas tiene una calidad. Si se suma de nuevo para todos los objetos, se obtiene:

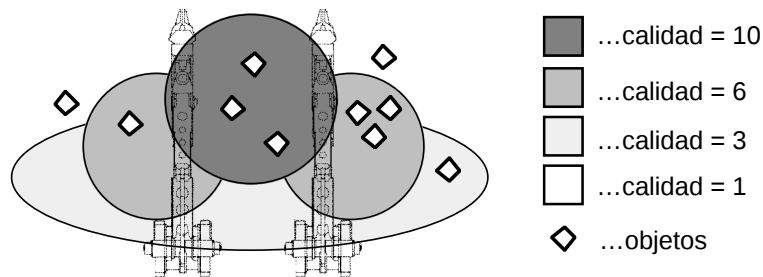
$$q_2 = \sum_k g_k \cdot q_{2k}$$

con

q_{2k} ...calidad de una zona espacial donde se encuentra el objeto k

g_k ...peso del objeto k

El ejemplo siguiente muestra zonas preferidas en forma de tres esferas interseccionadas (vista en planta)



Vista en planta con zonas preferidas

Para el ejemplo anterior se calcula el número de q_2 como sigue (todos los objetos con el peso 1):

$$\begin{aligned} q_2 &= \sum_{k=1}^{10} q_{2k} \\ &= 3 \cdot 10 + 4 \cdot 6 + 1 \cdot 3 + 2 \cdot 1 = 59 \end{aligned}$$

5.5.6.2. Criterios dependientes del poste

En el criterio dependiente del poste lo más importante es una colocación fuera de las zonas prohibidas. Sus datos geométricos son cargados, dependiendo del tipo del poste, de una base de datos o un fichero y están a disposición para cualquier cálculo. La formulación matemática es muy sencilla porque se trata de un criterio necesario de colocación. Ya que para este criterio todas las colocaciones fuera de las zonas prohibidas valen lo mismo, se puede dar la definición como sigue:

- Colocación dentro de las zonas prohibidas

$$q_3 = 0$$

- Colocación fuera de las zonas prohibidas

$$q_3 = 1$$

Esta evaluación tiene entonces prioridad absoluta, porque con una calidad de cero la colocación es eliminada. Aquí se ve que existen criterios necesarios y criterios suficientes. Que esta división coincida con la clasificación en dependiente e independiente, es una casualidad que resulta por el número pequeño de criterios.

5.5.7. Colocación final

Después del cálculo de la calidad de cada colocación la elección es muy sencilla. Se determina el máximo de la calidad y se consigue así la mejor colocación. En razón de la formulación aproximada de los criterios, podría tener sentido definir un intervalo de calidad en el que todas las colocaciones son igualmente favorecidas. Se puede proponer al operario un número manejable de posicionamientos del sistema, entre los cuales él elige manualmente la que le parece mejor. El gráfico siguiente corresponde a un ejemplo del apéndice B.6 y muestra el intervalo de las soluciones oscurecido.

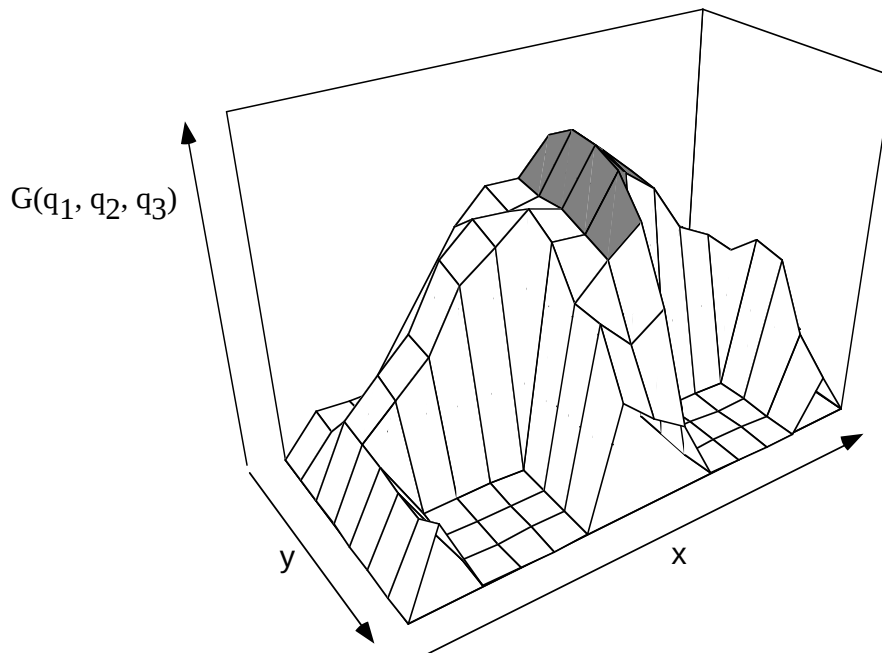


Gráfico de la calidad de colocaciones a las posiciones (x, y)

Las únicas aproximaciones que hemos hecho relativas al *objectlocation* han sido la clasificación del vector $\vec{a}$ en un intervalo de orientación y la representación del objeto en una trama. Se puede verificar la cantidad de soluciones que nos queda

todavía, con un gasto de tiempo justificable, usando las ecuaciones cinemáticas exactas.

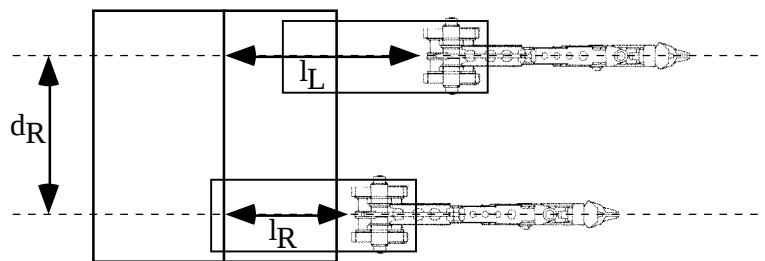
La elección de la colocación final la realiza ahora el usuario por si mismo. Para simplificarle esta tarea se le ofrece o un simulador o una visualización tridimensional simple. Con el simulador se pueden realizar dinámicamente las manipulaciones y por lo tanto no puede solo determinar la colocación final sino también hacer una planificación de trayectoria. Sin embargo, esta posibilidad está reservada, por el equipo y por el tiempo, al laboratorio.

Por contra la visualización tridimensional estática ofrece una herramienta sencilla para presentar al operario la colocación relativa al poste. El usuario entonces toma la decisión utilizando su experiencia anterior.

5.6. Posibilidades de ampliación

Debido a que el algoritmo se basa en una búsqueda secuencial de colocaciones posibles y buenas, resulta importante para aumentar la velocidad, realizar una exclusión de colocaciones que no puedan conducir a una solución. Ya ha sido hecho un estudio restringiendo el intervalo de la búsqueda, aunque muchos otros son posibles.

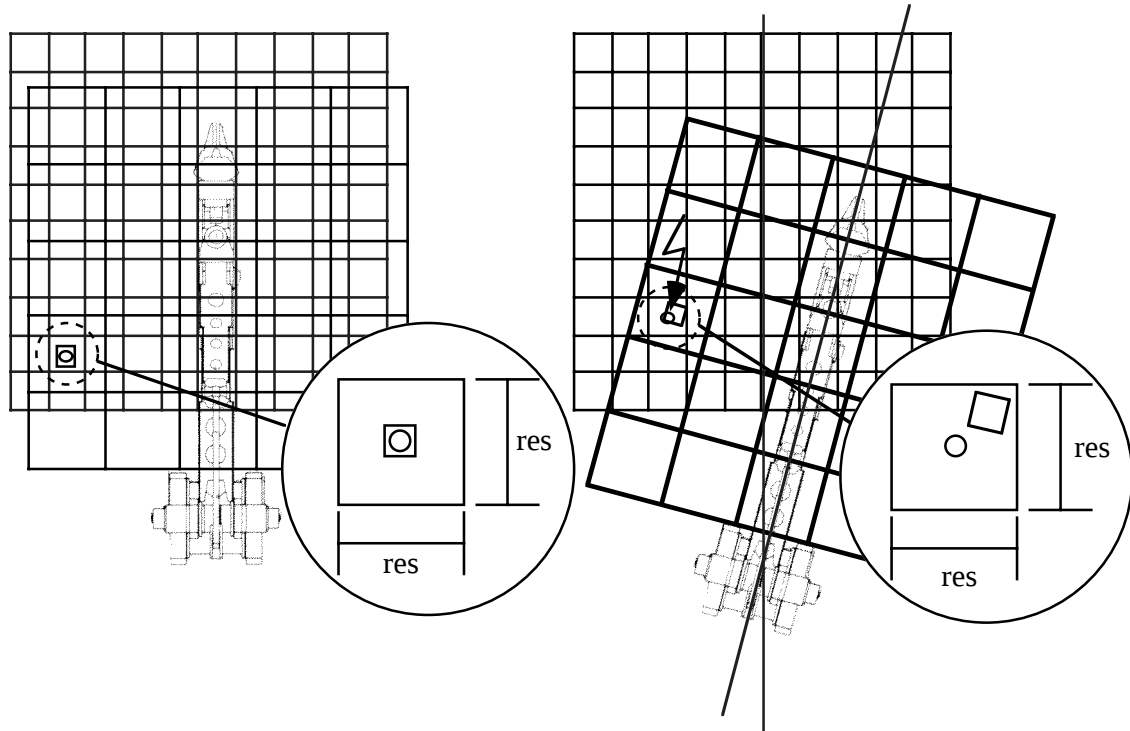
Debido a que se contempla separadamente la descripción del espacio de trabajo de cada robot, se puede introducir adicionalmente a d_R , más parámetros que describen la disposición del sistema. Si se realiza, por ejemplo, un sistema tal que los robots no son montados fijos en el eje x sino también varían en la dirección y (l_L , l_R), es muy fácil integrar este desplazamiento.



Ejemplo de parámetros para la plataforma

En general se puede implementar cada traslación (en la trama discreta dada por la resolución res) de los robots sin un cambio de su orientación. No se pueden ampliar

las consideraciones a giros en la disposición de los robots, porque las tramas no se pueden transformar mediante una multiplicación con enteros.

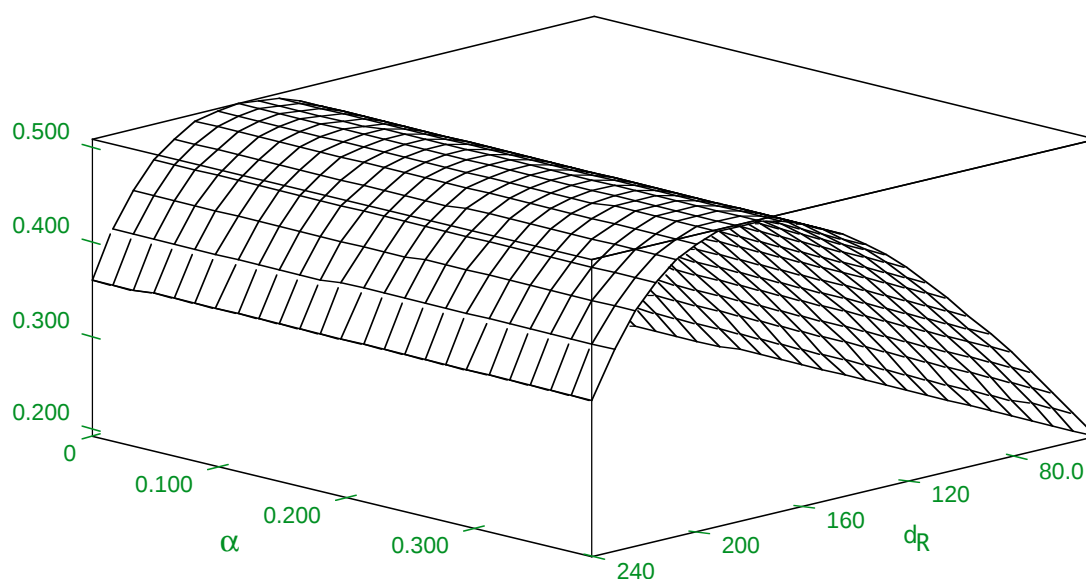


De momento el software sólo posee la posibilidad de buscar una colocación viable para los objetos definidos, si existe. Es decir, que el usuario tiene que cambiar el numero o la combinación de los objetos si el sistema no puede alcanzar todos los objetos sin mover el sistema. El software no hace propuestas de una combinación apropiada de los objetos. Sería posible, por la velocidad de la búsqueda de un posicionamiento, crear un algoritmo que parte todas las manipulaciones en subacciones que hagan que los cambios en la colocación del sistema sean los menos posibles.

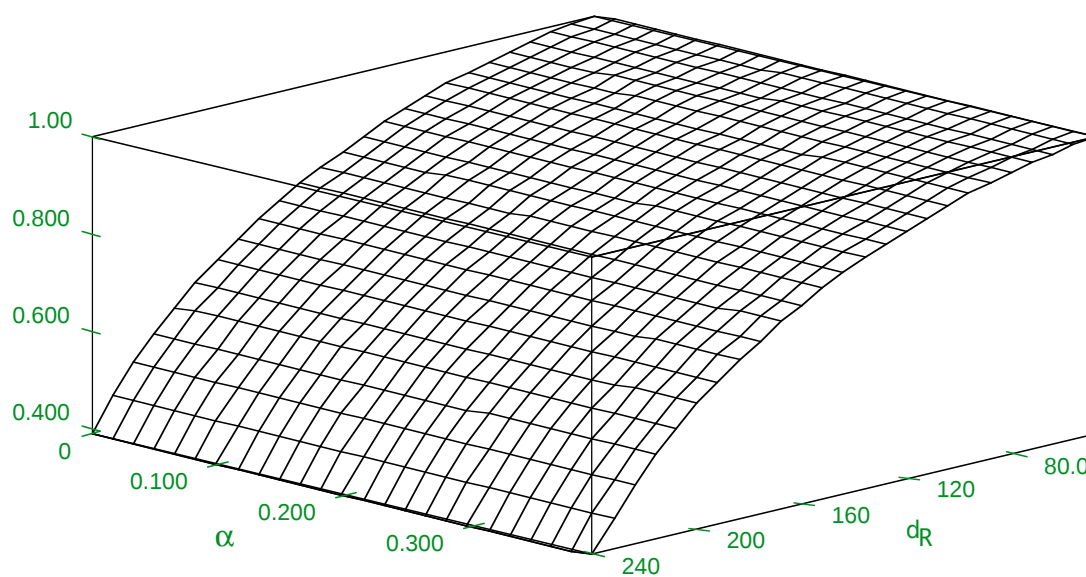
A. Apéndice A

A.1. Representación gráfica bidimensional de las medidas de evaluación dependientes de d_R y a

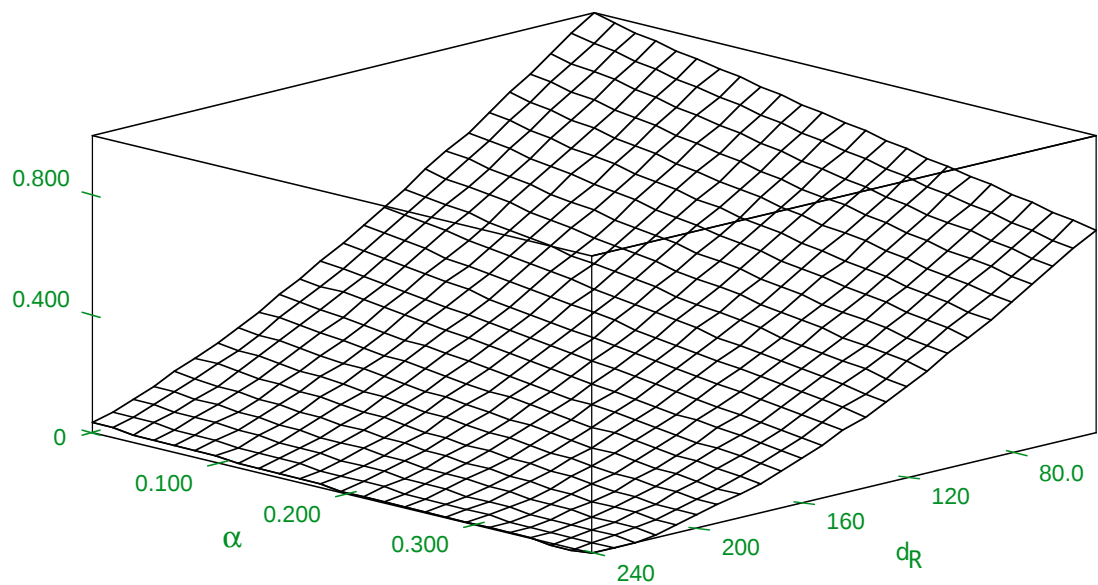
A.1.1 *Extensión máxima del agarre conjunto $d_{bypeff}(d_R, a)$*



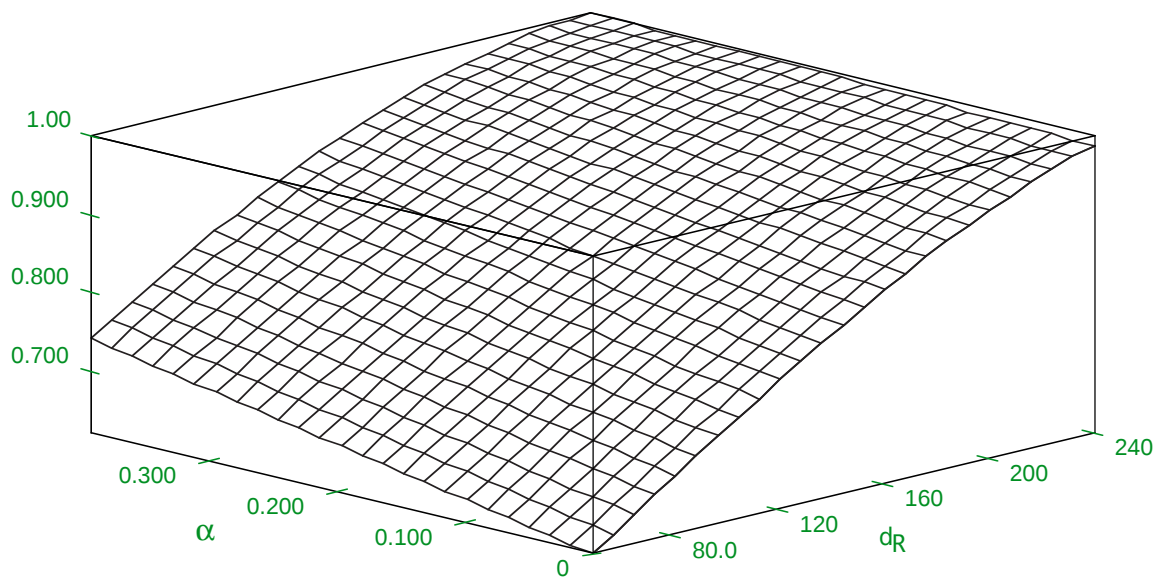
A.1.2 *Radio máximo de trabajo $r_{coordnorm}(d_R, a)$*



A.1.3 *Proyección en xy del espacio de trabajo común $F_{coordnorm}(d_R, a)$*



A.1.4 *Proyección en xy del espacio de trabajo total $F_{totnorm}(d_R, a)$*



A.2. Código de MAPLE para el cálculo de parámetros dR y a

```

with(plots):
printlevel:=1:
r := 130: drmin := 50:
drmax := 260:
alpha_max := Pi/8:

a1:=1: ## constante de peso para d_byp
a2:=1: ## constante de peso para r_coord
a3:=1: ## constante de peso para F_tot
a4:=1: ## constante de peso para F_coord

## intervalo valedero de las ecuaciones
drmax_plot := evalf(2*r*cos(alpha_max)):

## rcoord normalizada a rcoord a drmin
r_coord := sqrt(r^2-(dr/2)^2):
c_nrcoord := evalf(subs(dr=drmin,r_coord)):
r_coordnorm := r_coord/c_nrcoord:

## dbyp normalizada
d_byp := 2*dr:
c_ndbyp := 2*drmax:
d_bypnorm := d_byp/c_ndbyp:
d_bypeff := d_bypnorm*r_coordnorm:

## proyeccion del espacio de trabajo común normalizada a F_12
## con alpha=0 y dr = drmin
F_alpha0 := r^2*arccos(dr/(2*r)) - (dr/2)*r_coord:
F_sub1 := r^2*(alpha - cos(alpha)*sin(alpha)):

```

```

F_sub2 := (2*r*cos(alpha)-dr)*r*sin(alpha)-
(tan(alpha)*(0.5*(2*r*cos(alpha)-dr))^2):
F_coord := F_alpha0 - F_sub1 - F_sub2:
cnF_coord := evalf(subs(dr=drmin,F_alpha0)):
F_coordnorm := F_coord/cnF_coord:

## proyección del espacio total normalizada (F_12=0)
F_circulo := evalf(Pi) * r^2:
F_tot := F_circulo - F_coord:
F_totnorm := F_tot/F_circulo:

## Unificación a una función de calidad
## de la disposición
val := d_byeff^a1 * r_coordnorm^a2 * F_totnorm^a3 *
F_coordnorm^a4:

## plot tridimensional de la función
plot3d(val,dr=drmin..drmax_plot,alpha=0..alpha_max,axes=BOXED);

## determinación del valor de alpha al máximo
## (default = 0)
## con plot bidimensional
alpha_fixed := 0:
val_alpha := evalf(subs(alpha=alpha_fixed,opt)):
plot(opt_alpha,dr=drmin..drmax_plot);

## derivación de la función de calidad
opt := diff(val_alpha,dr):
plot(diff_opt,dr=drmin..drmax_plot);

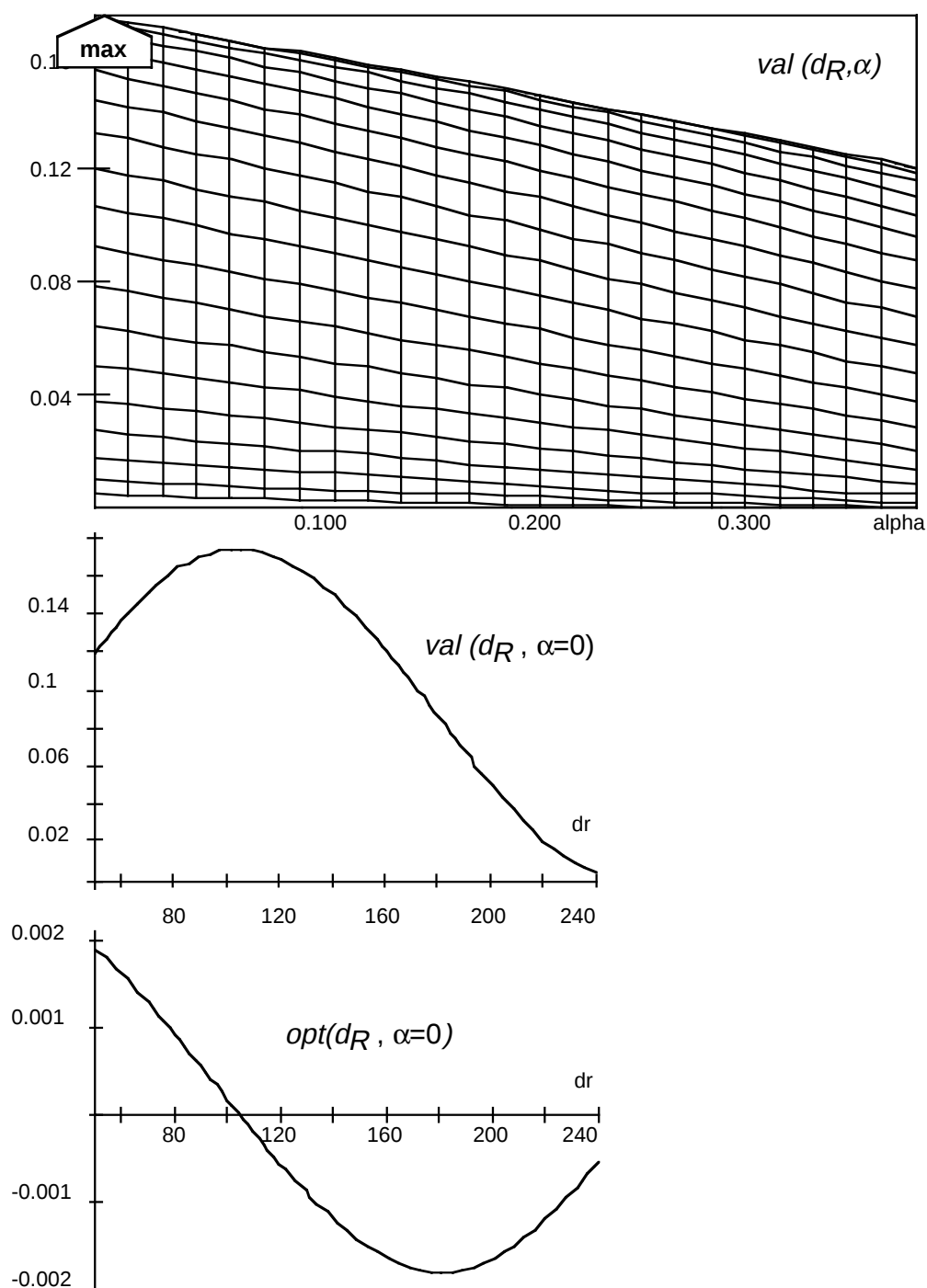
## solución de la ecuación val(dR) = 0
fsolve((diff_opt=0),dr,50..200);

```

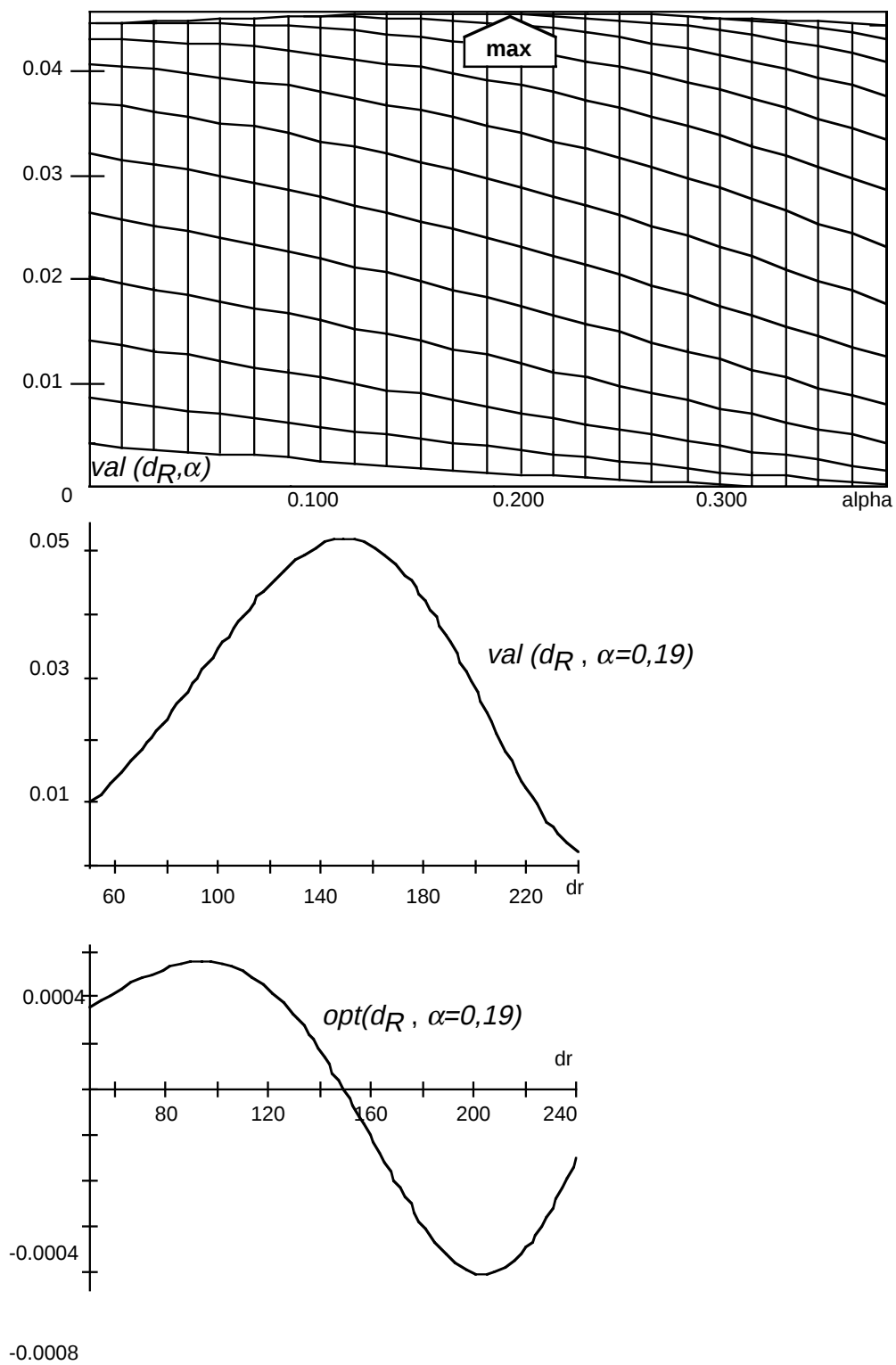
A.3. Ilustración gráfica de la determinación de d_R y a

Procedimiento práctico para la reducción de una función bidimensional a una unidimensional mediante el remplazo de a por un valor fijo de a máximo.

A.3.1 Constante de peso $k_3 = 1$



A.3.2 Constante de peso $k_3 = 8$



B. Apéndice B

B.1. Método según R.P.Paul para solucionar el problema cinemático inverso

Matrices de Denavit- Hartenberg y sus inversas para cada una de las seis articulaciones. S_n significa el $\sin Q_n$ y C_n el $\cos Q_n$.

$$A_1 = \begin{bmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_1^{-1} = \begin{bmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -S_1 & C_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} C_2 & -S_2 & 0 & C_2 a_2 \\ S_2 & C_2 & 0 & S_2 a_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2^{-1} = \begin{bmatrix} C_2 & S_2 & 0 & -a_2 \\ -S_2 & C_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} C_3 & -S_3 & 0 & C_3 a_3 \\ S_3 & C_3 & 0 & S_3 a_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^{-1} = \begin{bmatrix} C_3 & S_3 & 0 & a_3 \\ -S_3 & C_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} C_4 & 0 & S_4 & C_4 a_4 \\ S_4 & 0 & -C_4 & S_4 a_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4^{-1} = \begin{bmatrix} C_4 & S_4 & 0 & -a_4 \\ 0 & 0 & 1 & 0 \\ S_4 & -C_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5 = \begin{bmatrix} C_5 & 0 & -S_5 & 0 \\ S_5 & 0 & C_5 & 0 \\ 0 & -1 & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5^{-1} = \begin{bmatrix} C_5 & S_5 & 0 & 0 \\ 0 & 0 & -1 & d_5 \\ -S_5 & C_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_6 = \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_6^{-1} = \begin{bmatrix} C_6 & S_6 & 0 & 0 \\ -S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformación directa mediante la multiplicación de las matrices A_1 hasta A_6 .

$$U_6 = A_6$$

$$U_5 = A_5 U_6 = \begin{bmatrix} C_5 C_6 & -C_5 S_6 & -S_5 & 0 \\ S_5 C_6 & -S_5 S_6 & C_5 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$U_4 = A_4 U_5 = \begin{bmatrix} C_4 C_5 C_6 - S_4 S_6 & -C_4 C_5 S_6 - S_4 C_6 & -C_4 S_5 & S_4 d_5 + C_4 a_4 \\ S_4 C_5 C_6 + C_4 S_6 & -S_4 C_5 S_6 + C_4 C_6 & -S_4 S_5 & -C_4 d_5 + S_4 a_4 \\ S_5 C_6 & -S_5 S_6 & C_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$U_3 = A_3 U_4 = \begin{bmatrix} X_1 C_3 + X_2 S_3 & X_3 C_3 + X_4 S_3 & -C_3 C_4 S_5 + S_3 S_4 S_5 \\ -X_2 C_3 + X_1 S_3 & -X_4 C_3 + X_3 S_3 & -S_3 C_4 S_5 - C_3 S_4 S_5 \\ S_5 C_6 & -S_5 S_6 & C_5 \\ 0 & 0 & 0 \\ C_3(a_4 C_4 + S_4 d_5 + a_3) + S_3(C_4 d_5 - a_4 S_4) \\ C_3(-C_4 d_5 + a_4 S_4) + S_3(a_4 C_4 + S_4 d_5 + a_3) \\ 0 \\ 1 \end{bmatrix}$$

$$U_2 = A_2 U_3 = \begin{bmatrix} X_2 S_{23} + X_1 C_{23} & X_4 S_{23} + X_3 C_{23} & S_{23} S_4 S_5 - C_{23} C_4 S_5 \\ X_1 S_{23} - X_2 C_{23} & X_3 S_{23} - X_4 C_{23} & -S_{23} S_5 C_4 - C_{23} S_4 S_5 \\ S_5 C_6 & -S_5 S_6 & C_5 \\ 0 & 0 & 0 \\ C_2 a_2 + C_{23}(a_4 C_4 + S_4 d_5 + a_3) + S_{23}(C_4 d_5 - a_4 S_4) \\ S_2 a_2 + S_{23}(a_4 C_4 + S_4 d_5 + a_3) + C_{23}(-C_4 d_5 + a_4 S_4) \\ 0 \\ 1 \end{bmatrix}$$

$$U_1 = A_1 U_2 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

con

$$X_1 = C_4 C_5 C_6 - S_4 S_6$$

$$X_2 = -S_4 C_5 C_6 - C_4 S_6$$

$$X_3 = -C_4 C_5 S_6 - C_4 C_6$$

$$X_4 = S_4 C_5 S_6 - C_4 C_6$$

y con

$$n_x = C_1(X_2 S_{23} + X_1 C_{23}) - S_1 S_5 C_6$$

$$n_y = S_1(X_2 S_{23} + X_1 C_{23}) + C_1 S_5 C_6$$

$$n_z = -X_1 S_{23} + X_2 C_{23}$$

$$o_x = C_1(X_4 S_{23} + X_3 C_{23}) + S_1 S_5 S_6$$

$$o_y = S_1(X_4 S_{23} + X_3 C_{23}) - C_1 S_5 S_6$$

$$o_z = -X_3 S_{23} + X_4 C_{23}$$

$$a_x = C_1(S_{23} S_4 S_5 - C_{23} C_4 S_5) - S_1 C_5$$

$$a_y = S_1(S_{23} S_4 S_5 - C_{23} C_4 S_5) + C_1 C_5$$

$$a_z = S_{23} S_5 C_4 + C_{23} S_4 S_5$$

$$p_x = C_1(C_2 a_2 + C_{23} a_4 C_4 + C_{23} S_4 d_5 + C_{23} a_3 + S_{23} C_4 d_5 - S_{23} a_4 S_4)$$

$$p_y = S_1(C_2 a_2 + C_{23} a_4 C_4 + C_{23} S_4 d_5 + C_{23} a_3 + S_{23} C_4 d_5 - S_{23} a_4 S_4)$$

$$p_z = -S_2 a_2 - S_{23} a_4 C_4 - S_{23} S_4 d_5 - S_{23} a_3 + C_{23} C_4 d_5 - C_{23} a_4 S_4$$

Determinación de las ecuaciones cinemáticas inversas mediante una multiplicación sucesiva y una determinación de ángulos o sumas de ángulos utilizando los ángulos conocidos.

$$U_2 = A_1^{-1}T_6 = \begin{bmatrix} f_{11}(n) & f_{11}(0) & f_{11}(a) & f_{11}(p) \\ f_{12}(n) & f_{12}(0) & f_{12}(a) & f_{12}(p) \\ f_{13}(n) & f_{13}(0) & f_{13}(a) & f_{13}(p) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

con

$$f_{11} = C_1x + S_1y$$

$$f_{12} = -z$$

$$f_{13} = -S_1x + C_1y$$

$$f_{13}(p) = 0$$

$\Rightarrow$

$$-S_1p_x + C_1p_y = 0$$

con

$$p_x = r \cos \phi; p_y = r \sin \phi$$

$$\Rightarrow \phi = \arctan\left(\frac{p_y}{p_x}\right); r = \sqrt{p_x^2 + p_y^2}$$

$$\sin(\phi - \Theta_1) = 0$$

$$\Rightarrow \Theta_1 = \arctan\left(\frac{p_y}{p_x}\right)$$

$$f_{11}(a) = C_1a_x + S_1a_y = S_5(S_{23}S_4 - C_{23}C_4) = S_5(-C_{234})$$

$$f_{12}(a) = -a_z = S_5(-S_{23}C_4 - C_{23}S_4) = S_5(-S_{234})$$

$$\Rightarrow \tan(\Theta_2 + \Theta_3 + \Theta_4) = \frac{f_{12}(a)}{f_{11}(a)} = \frac{-a_z}{C_1a_x + S_1a_y}$$

$$\Rightarrow (\Theta_2 + \Theta_3 + \Theta_4) = \arctan\left(\frac{a_z}{C_1a_x + S_1a_y}\right)$$

$$f_{11}(p) = C_1 p_x + S_1 p_y = C_2 a_2 + C_{23}(a_4 C_4 + S_4 d_5 + a_3) + S_{23}(C_4 d_5 - a_4 S_4)$$

$$f_{12}(p) = -p_z = S_2 a_2 + S_{23}(a_4 C_4 + S_4 d_5 + a_3) + C_{23}(-C_4 d_5 + a_4 S_4)$$

$\Rightarrow$

$$f_{11}(p) = C_2 a_2 + a_4 C_{234} + d_5 S_{234} + a_3 C_{23}$$

$$f_{12}(p) = S_2 a_2 + a_4 S_{234} - d_5 C_{234} + a_3 S_{23}$$

con

$$u = a_4 C_{234} + d_5 S_{234} = \text{const.}$$

$$v = a_4 S_{234} - d_5 C_{234} = \text{const.}$$

$\Rightarrow$

$$f_{11p} = C_2 a_2 + a_3 C_{23} + u$$

$$f_{12p} = S_2 a_2 + a_3 S_{23} + v$$

elevando al cuadrado

$\Rightarrow$

$$(C_2 a_2)^2 + 2a_2 a_3 C_2 C_{23} + (a_3 C_{23})^2 = (f_{11p} - u)^2$$

$$(S_2 a_2)^2 + 2a_2 a_3 S_2 S_{23} + (a_3 S_{23})^2 = (f_{12p} - v)^2$$

sumando

$\Rightarrow$

$$a_2^2 + a_3^2 + 2a_2 a_3 (C_{23} C_2 + S_{23} S_2) = (f_{11p} - u)^2 + (f_{12p} - v)^2$$

con

$$C_{23} C_2 + S_{23} S_2 = C_2 C_3 C_2 - S_2 S_3 C_2 + S_2 C_3 S_2 + C_2 S_3 S_2 = C_3 (C_2^2 + S_2^2) = C_3$$

$$\Rightarrow \Theta_3 = \arccos \left(\frac{(C_1 p_x + S_1 p_y - a_4 C_{234} - d_5 S_{234})^2 + (-p_z - a_4 S_{234} + d_5 C_{234})^2 - a_2^2 - a_3^2}{2a_2 a_3} \right)$$

$$U_4 = A_3^{-1} A_2^{-1} A_1^{-1} T_6 = \begin{bmatrix} f_{31}(n) & f_{31}(0) & f_{31}(a) & f_{31}(p) - a_2 C_3 - a_3 \\ f_{32}(n) & f_{32}(0) & f_{32}(a) & f_{32}(p) + a_2 S_3 \\ f_{33}(n) & f_{33}(0) & f_{33}(a) & f_{33}(p) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

con

$$f_{31} = C_{23} f_{11} + S_{23} f_{12}$$

$$f_{32} = -S_{23} f_{11} + C_{23} f_{12}$$

$$f_{33} = f_{13}$$

$$S_4 d_5 + C_4 a_4 = C_{23} f_{11p} + S_{23} f_{12p} - a_2 C_3 - a_2$$

$$-C_4 d_5 + S_4 a_4 = C_{23} f_{12p} - S_{23} f_{11p} + a_2 S_3$$

con

$$k = a_2 C_3 + a_3$$

$$l = a_2 S_3$$

elevado al cuadrado

$$(S_4 d_5)^2 + 2S_4 d_5 C_4 a_4 + (C_4 a_4)^2 = (C_{23} f_{11p} + S_{23} f_{12p} - k)^2$$

$$(C_4 d_5)^2 - 2C_4 d_5 S_4 a_4 + (S_4 a_4)^2 = (C_{23} f_{12p} - S_{23} f_{11p} + l)^2$$

sumando

$$\begin{aligned} a_4^2 + d_5^2 &= [(C_{23} f_{11p})^2 + 2C_{23} S_{23} f_{11p} f_{12p} + (S_{23} f_{12p})^2] - 2k(C_{23} f_{11p} + S_{23} f_{12p}) + k^2 \\ &\quad + [(C_{23} f_{12p})^2 - 2C_{23} S_{23} f_{11p} f_{12p} + (S_{23} f_{11p})^2] + 2l(C_{23} f_{12p} - S_{23} f_{11p}) + l^2 \\ &= f_{11p}^2 + f_{12p}^2 + k^2 + l^2 + C_{23}(2lf_{12p} - 2kf_{11p}) + S_{23}(-2lf_{11p} - 2kf_{12p}) \end{aligned}$$

con

$$e = 2lf_{12p} - 2kf_{11p}$$

$$f = -2lf_{11p} - 2kf_{12p}$$

$$d = a_4^2 + d_5^2 - (f_{11p}^2 + f_{12p}^2 + k^2 + l^2)$$

$$\Rightarrow d = eC_{23} + fS_{23}$$

$$\text{con } f = r \cos \phi ; \quad e = r \sin \phi$$

$$\phi = \arctan\left(\frac{e}{f}\right) ; \quad r = \sqrt{e^2 + f^2}$$

$$\sin(\phi + \Theta_{23}) = \frac{d}{r} \quad \cos(\phi + \Theta_{23}) = \pm \sqrt{1 - \left(\frac{d}{r}\right)^2}$$

$$\Rightarrow \Theta_{23} = -\arctan\left(\frac{e}{f}\right) + \arctan\left(\frac{d}{-\sqrt{e^2 + f^2 - d^2}}\right)$$

$$\Rightarrow \Theta_4 = \Theta_{234} - \Theta_{23}$$

$$\Rightarrow \Theta_2 = \Theta_{23} - \Theta_3$$

$$U_5 = A_4^{-1} A_3^{-1} A_2^{-1} A_1^{-1} T_6 = \begin{bmatrix} f_{41}(n) & f_{41}(0) & f_{41}(a) & f_{41}(p) + a_2(C_{34} - C_4) - a_4 \\ f_{42}(n) & f_{42}(0) & f_{42}(a) & f_{42}(p) \\ f_{43}(n) & f_{43}(0) & f_{43}(a) & f_{43}(p) + a_2(S_{34} - S_4) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

con

$$f_{41} = C_4 f_{31} + S_4 f_{32}$$

$$f_{42} = f_{33}$$

$$f_{43} = S_4 f_{31} - C_4 f_{32}$$

$$f_{41a} = -S_5$$

$$f_{42a} = C_5$$

$$\begin{aligned} \Rightarrow \tan \Theta_5 &= \frac{-f_{41a}}{f_{42a}} = -\frac{C_4 f_{31a} + S_4 f_{32a}}{f_{33a}} \\ &= -\frac{C_4(C_{23} f_{11a} + S_{23} f_{12a}) + S_4(-S_{23} f_{11a} + C_{23} f_{12a})}{f_{13a}} \end{aligned}$$

$$\Rightarrow \Theta_5 = \arctan \left(\frac{C_4(C_{23}[C_1 a_x S_1 a_y] + S_{23}[-a_z]) + S_4(-S_{23}[C_1 a_x S_1 a_y] + C_{23}[-a_z])}{-S_1 a_x + C_1 a_y} \right)$$

$$f_{41n} = C_5 C_6$$

$$f_{41o} = -C_5 S_6$$

$$\Rightarrow \tan \Theta_6 = \frac{-f_{41o}}{f_{41n}} = -\frac{C_4 f_{31o} + S_4 f_{32o}}{C_4 f_{31n} + S_4 f_{32n}}$$

$$\Rightarrow \Theta_6 = \arctan \left(\frac{C_4(C_{23}[C_1 o_x S_1 o_y] + S_{23}[-o_z]) + S_4(-S_{23}[C_1 o_x S_1 a_y] + C_{23}[-o_z])}{C_4(C_{23}[C_1 n_x S_1 n_y] + S_{23}[-n_z]) + S_4(-S_{23}[C_1 n_x S_1 n_y] + C_{23}[-n_z])} \right)$$

B.2. Código en C de las ecuaciones cinemáticas inversas para el simulador TOROS

```
/******  
cinInvKraft(r, pos, or, conf, velFlg)  
Robot *r;  
punto3d pos, or;  
int conf, velFlg;  
/{*****  
int i;  
float DH[4][4], ang[6];  
double f11a, f11p, f12p, f12a, f13a, f13n, f13o, f31a, f32a, f41a, f42a,  
       f11o, f11n, f12o, f12n, f31o, f31n, f32o, f32n, f41o, f41n,  
       u, v, w, k, l, d, e, f, ang23, ang234, ang23_1, ang23_2, t_val;  
static int inic=0;  
static float a2, a3, a4, d5;  
  
if (!inic) {  
    a2=r->modCin[1][3][0];  
    a3=r->modCin[2][3][0];  
    a4=r->modCin[3][3][0];  
    d5=r->modCin[4][3][2];  
    inic=1;  
}  
getDH(r, pos, or, DH);  
  
/****** Teta 1 *****/  
if (DH[3][0] <= 0)  
    ang[0]=-arcTag(DH[3][1], -DH[3][0]);  
else  
    ang[0]=arcTag(DH[3][1], DH[3][0]);
```

```

if (verifLim(r, 0, ang, velFlg) != PACA_OK)
    return NO_SOLUCION;

f11a=cos(ang[0])*DH[2][0]+sin(ang[0])*DH[2][1];
f12a= -DH[2][2];
f13a= -sin(ang[0])*DH[2][0]+cos(ang[0])*DH[2][1];
f11p=cos(ang[0])*DH[3][0]+sin(ang[0])*DH[3][1];
f12p= -DH[3][2];
f11o=cos(ang[0])*DH[1][0]+sin(ang[0])*DH[1][1];
f11n=cos(ang[0])*DH[0][0]+sin(ang[0])*DH[0][1];
f12o= -DH[1][2];
f12n= -DH[0][2];

ang234=arcTag(f12a,f11a);

u=a4*cos(ang234)+d5*sin(ang234);
v=a4*sin(ang234)-d5*cos(ang234);

/***** Teta 3 *****/

w = ((SQR(f11p-u)+SQR(f12p-v))-SQR(a2)-SQR(a3))/(2*a2*a3);
if (fabs(w) > 1)
    w = 1.0;
ang[2]=arcTag(sqrt(1-SQR(w)),w);

if (verifLim(r, 2, ang, velFlg) != PACA_OK)
    return NO_SOLUCION;

k=a2*cos(ang[2])+a3;
l=a2*sin(ang[2]);
e=2*(l*f12p-k*f11p);
f= -2*(l*f11p+k*f12p);
d=SQR(a4)+SQR(d5)-(SQR(f11p)+SQR(f12p)+SQR(k)+SQR(l));

```

```

if (SQR(e)+SQR(f) < SQR(d)) {
    sprintf(torosError, "KRAFT: Punto fuera del alcance del robot\n");
    return NO_SOLUCION;
}

ang23_1=-arcTag(e,f)+arcTag(d, -sqrt(SQR(e)+SQR(f)-SQR(d)));
ang23_2=arcTag(e,-f)-arcTag(d, -sqrt(SQR(e)+SQR(f)-SQR(d)));

t_val=cos(ang23_1-ang[2])*a2+a4*cos(ang234)+d5*sin(ang234)+a3*cos(ang23_1);

if (IGUAL(t_val, f11p))
    ang23 = ang23_1;
else
    ang23 = ang23_2;

/***** Teta 4 *****/
ang[3]=ang234 - ang23;

if (verifLim(r, 3, ang, velFlg) != PACA_OK)
    return NO_SOLUCION;

/***** Teta 2 *****/
ang[1]=ang23 - ang[2];

if (verifLim(r, 1, ang, velFlg) != PACA_OK)
    return NO_SOLUCION;

f32a= -sin(ang23)*f11a + cos(ang23)*f12a;
f31a=cos(ang23)*f11a + sin(ang23)*f12a;
f42a=f13a;
f41a=cos(ang[3])*f31a + sin(ang[3])*f32a;

```

```

/***** Teta 5 *****/
ang[4]=arcTag(-f41a,f42a);

if (verifLim(r, 4, ang, velFlg) != PACA_OK)
    return NO_SOLUCION;

/***** Teta 6 *****/

f31o=cos(ang23)*f11o + sin(ang23)*f12o;
f31n=cos(ang23)*f11n + sin(ang23)*f12n;
f32o= -sin(ang23)*f11o + cos(ang23)*f12o;
f32n= -sin(ang23)*f11n + cos(ang23)*f12n;
f41o=cos(ang[3])*f31o + sin(ang[3])*f32o;
f41n=cos(ang[3])*f31n + sin(ang[3])*f32n;

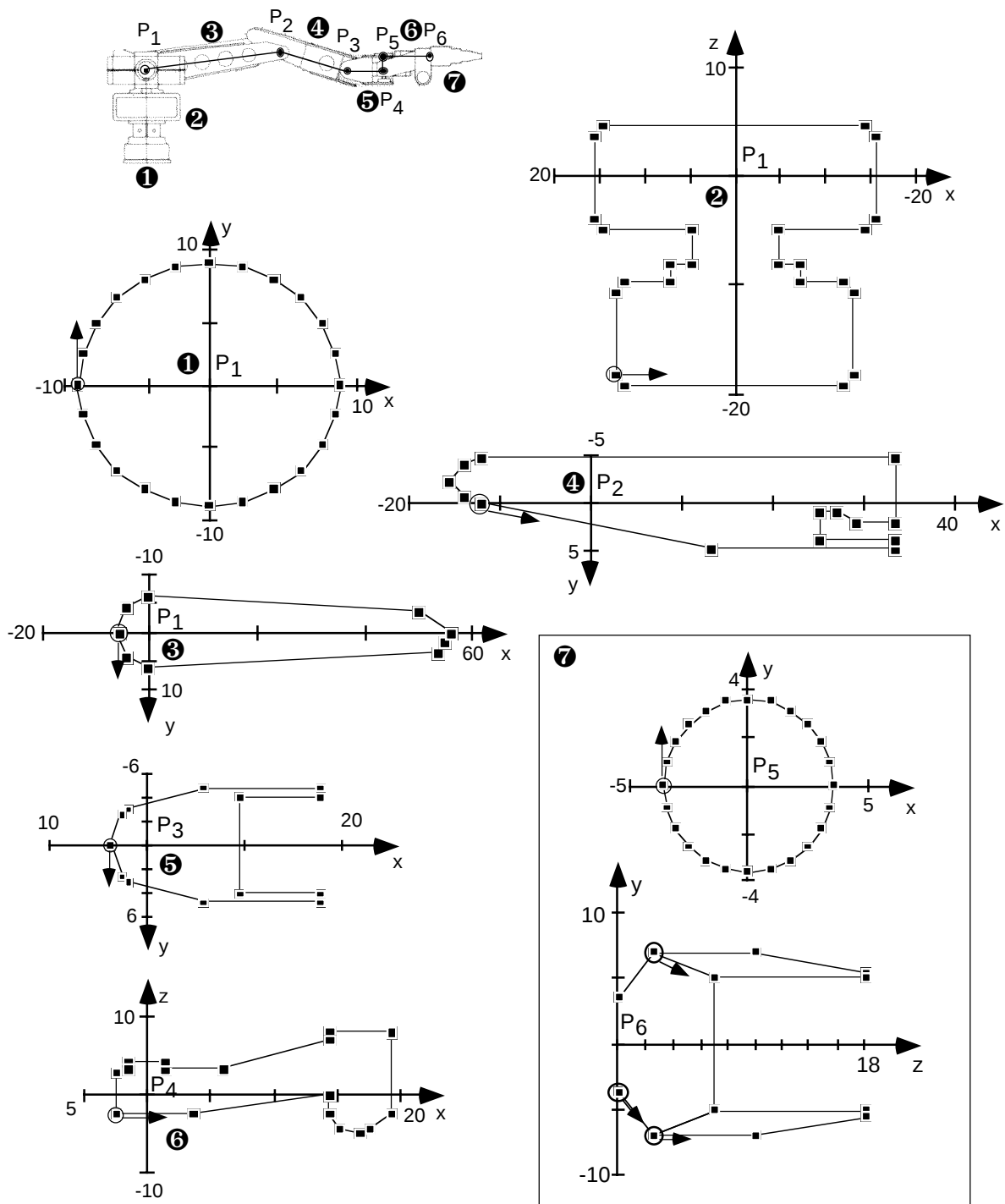
if (IGUAL(cos(ang[4]), 0.0))
    ang[5]=r->valArt[5]*r->relArt[5][0]+r->relArt[5][1];
else if (cos(ang[4]) > 0)
    ang[5]=arcTag(-f41o,f41n);
else if (cos(ang[4]) < 0)
    ang[5]=arcTag(f41o,-f41n);

if (verifLim(r, 5, ang, velFlg) != PACA_OK)
    return NO_SOLUCION;

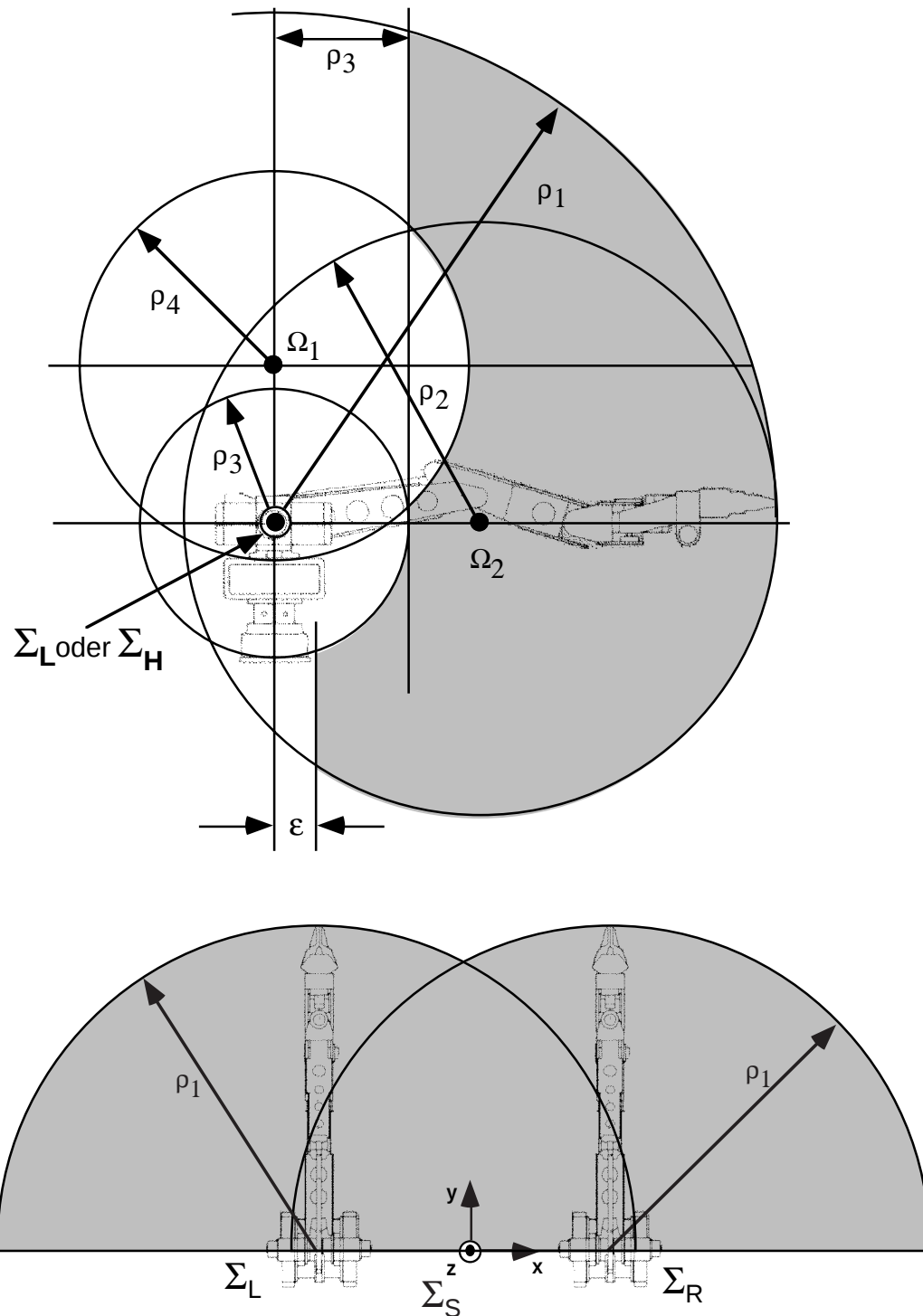
for (i=0; i < 6; i++) {
    r->valArt[i]=(ang[i]-r->relArt[i][1])/r->relArt[i][0];
    /*
        printf("ang %d = %f\n", i+1, valArt[i] rad);
    */
}
return PACA_OK;

```

B.3. Definición de los elementos mecánicos del robot GRIPS en TOROS



B.4. Descripción de las proyecciones del espacio de trabajo por círculos y líneas

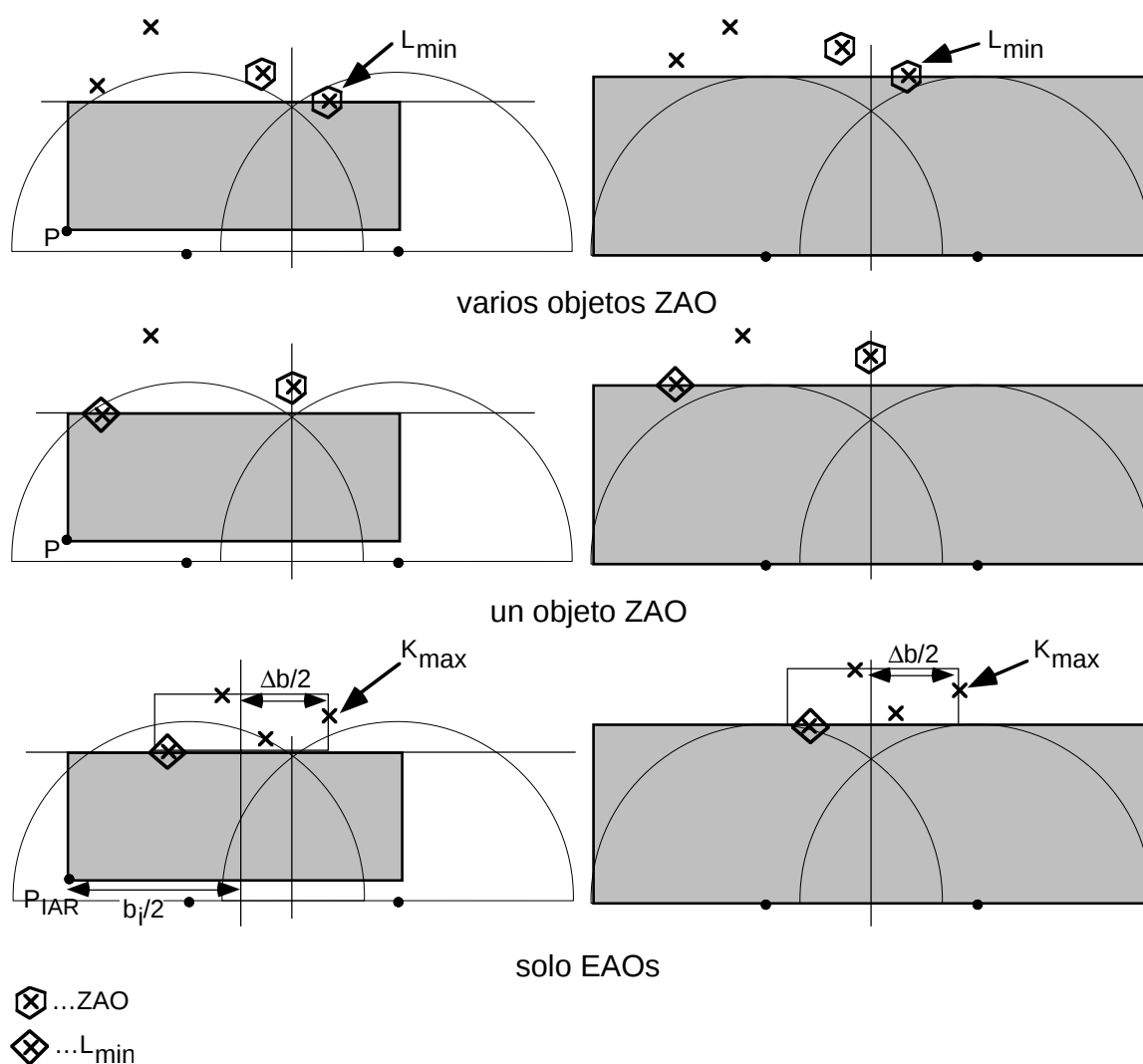


B.5. Determinación de posiciones iniciales x_a e y_a

La siguiente tabla muestra gráficamente los criterios para la determinación de la colocación inicial dentro de la prueba de plausibilidad en los casos de una o ninguna restricción del espacio de trabajo.

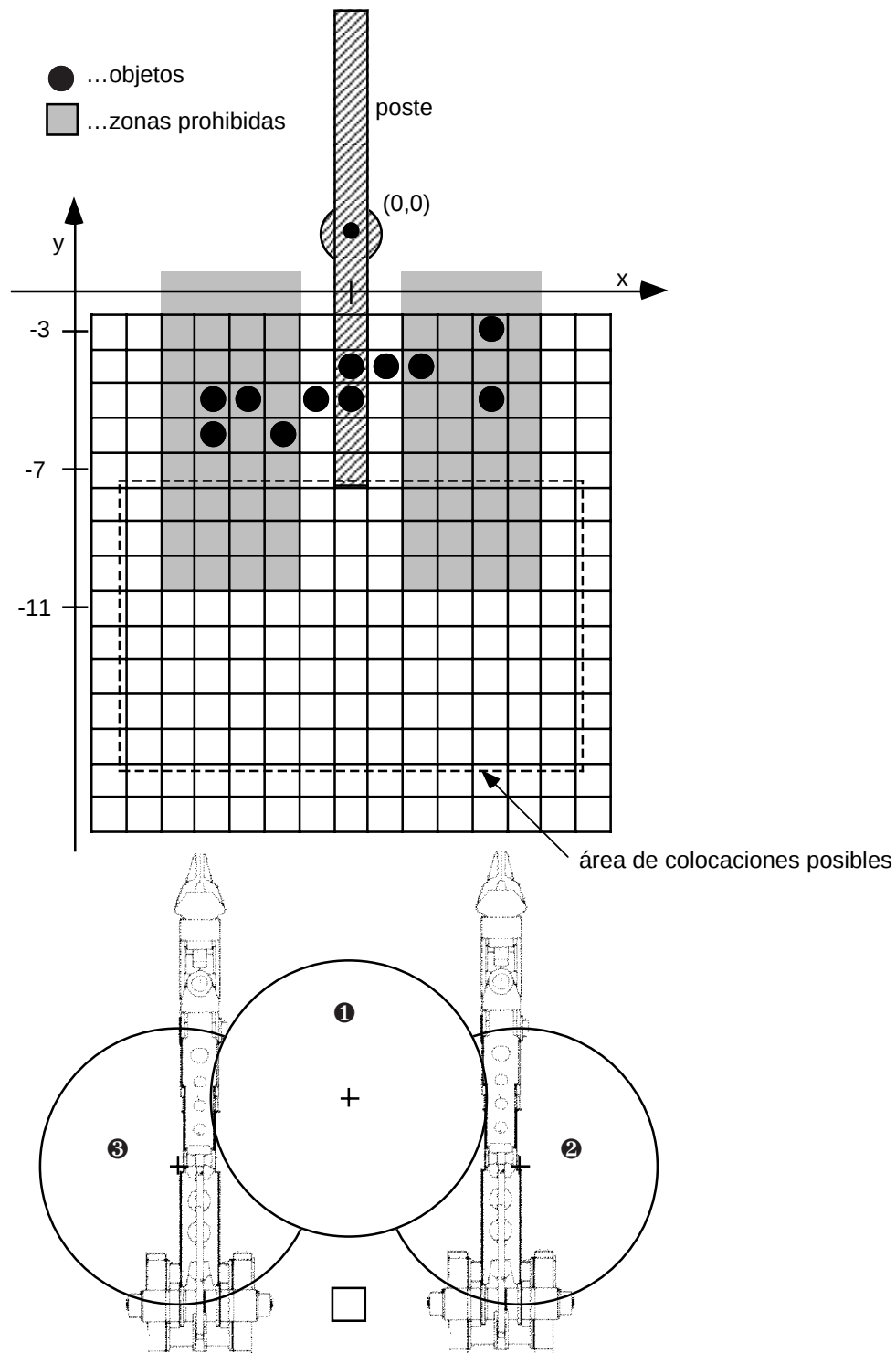
Restricción del espacio de trabajo

Ninguna restricción del espacio de trabajo



B.6. Ejemplo de evaluación de una colocación

B.6.1 Definición de los objetos y de las áreas preferidas del sistema



B.6.2 Programa MAPLE para el cálculo de la evaluación del ejemplo

```

with(plots):
readlib(write):
open(out.dat):

## definición de los parámetros ##
num_obj := 11:
nu1 := 0.5:
nu2 := 1:
g1 := 5:
g2 := 5:
g3 := 5:

## definición de los objetos ##
O[1] := -4,-6:
O[2] := -4,-5:
O[3] := -3,-5:
O[4] := -2,-6:
O[5] := -1,-5:
O[6] := 0,-5:
O[7] := 0,-4:
O[8] := 1,-4:
O[9] := 2,-4:
O[10] := 4,-3:
O[11] := 4,-5:
plotlist := NULL:

for y from -15 to -7 do
    excellist := NULL:
    for x from -6 to 6 do

```

```
##### prueba a zonas prohibidas (criterio 3) #####
q3 := 1:

### zona 1 ###
if (x>(-6)) and (x<(-1)) and (y>(-11)) then q3 := 0 fi;

### zona 2 ###
if (x<6) and (x>1) and (y>(-11)) then q3 := 0 fi;

#####

##### criterio 1 (distancia de los objetos) #####
q1 := 0;

for o from 1 to num_obj do
  q1h := q1:
  q1 := q1h + sqrt( (x - O[o][1])^2 + (y - O[o][2])^2 ):
od;

#####

##### criterio 2 (áreas preferidas) #####
q2 := 1;

## área con calidad g1 (circulo con radio 4 MP1(0,6)) ##
a := 0 : b := 6:
rad1 := 4:
for o from 1 to num_obj do
  q2h := q2;
  r1 := sqrt( ((x+a) - O[o][1])^2 + ((y+b) - O[o][2])^2 ):
  if (evalf(r1) < rad1) then q2 := q2h + g1 fi:
od;

## área con calidad g2 (circulo con radio 4 MP2(6,3)) ##
a := 6 : b := 3:
rad2 := 4:
```

```

for o from 1 to num_obj do
    q2h := q2;
    r2 := sqrt( ((x+a) - O[o][1])^2 + ((y+b) - O[o][2])^2 );
    if (evalf(r2) < rad2) and not(evalf(r1) < rad1)
        then q2 := q2h + q2 fi;
od;

## área con calidad g3 (circulo con radio 4 ,MP3(-6,3)) ##
a := -6 : b := 3:
rad3 := 4:
for o from 1 to num_obj do
    q2h := q2;
    r3 := sqrt( ((x+a) - O[o][1])^2 + ((y+b) - O[o][2])^2 );
    if (evalf(r3) < rad3) and not(evalf(r1) < rad1)
        then q2 := q2h + g3 fi;
od;

G := evalf(q1^nu1 * q2^nu2 * q3);

#####
if (G<>0) then plotlist := plotlist,[x,y,G] fi;
excellist := excellist,trunc(G):

od;
writeln(excellist):
od;

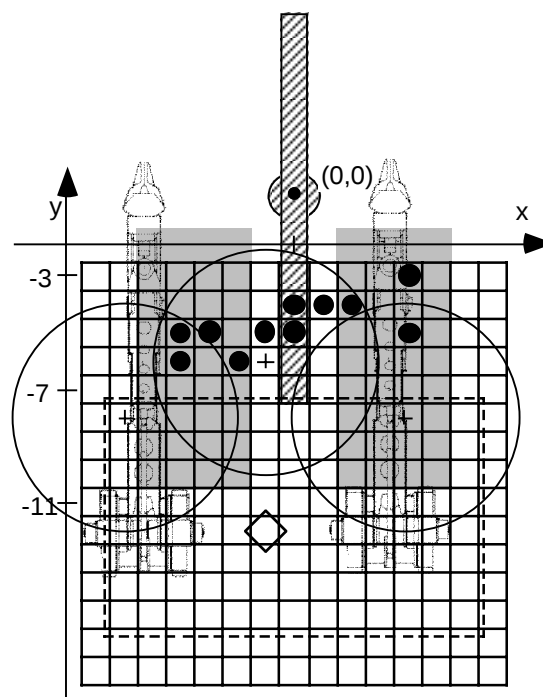
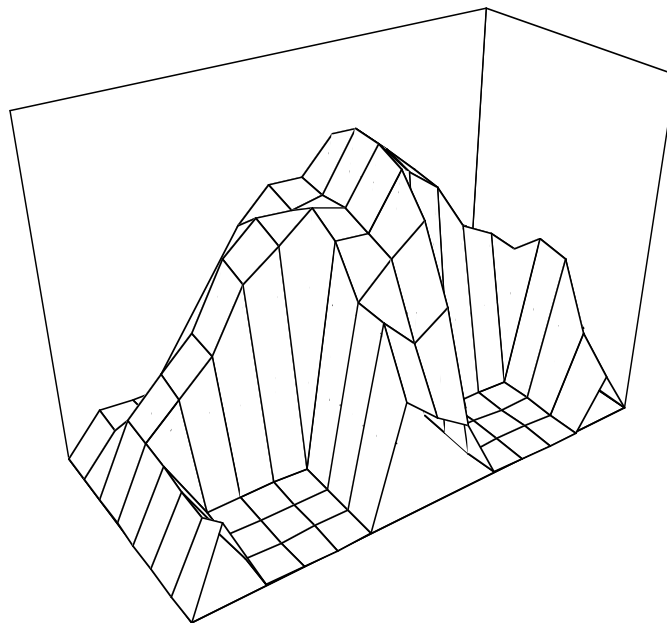
## visualización de los datos ##
pointplot({plotlist},axes=BOXED);

close(out.dat):

```

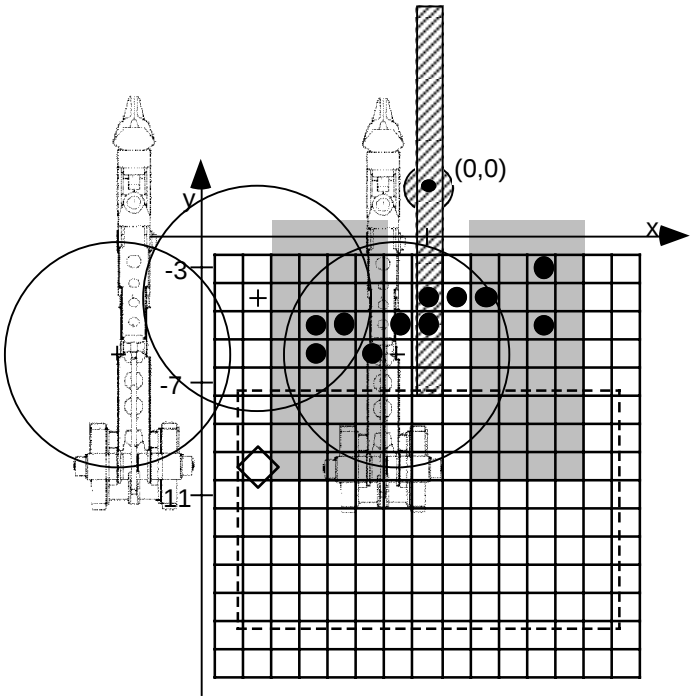
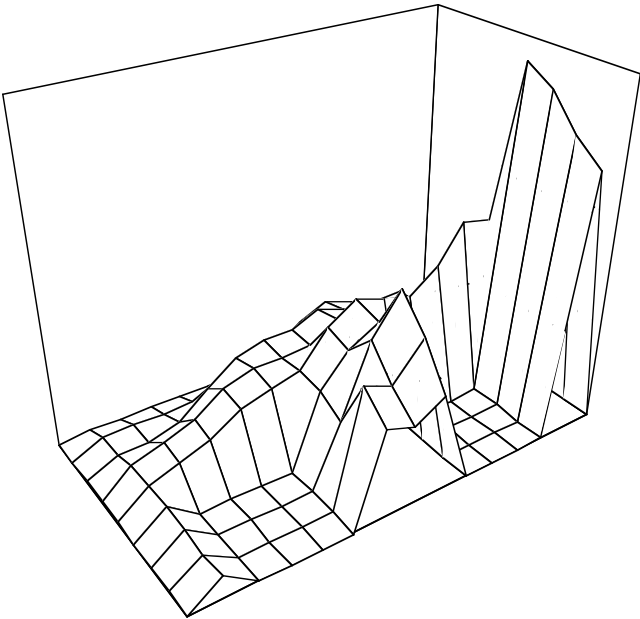
B.6.3 Alternativa 1

peso de los criterios		calidad de las zonas pref. g_i			mejor colocación
n_1	n_2	g_1	g_2	g_3	(x,y)
0,5	1	20	5	5	(1,-12)



B.6.4 Alternativa 2

peso de los criterios		calidad de las zonas pref. g_i			mejor colocación
n_1	n_2	g_1	g_2	g_3	(x,y)
0,5	1	5	20	1	(-6,-10)



B.7. Código de MAPLE para el cálculo del espacio de trabajo discreto

wsnum := proc():

-- CÁLCULO DEL ESPACIO DE TRABAJO PARA --

-- LA VISUALISACIÓN CON AUTOCAD --

--- leer biblioteca ---

readlib(write):

with(plots):

--- abrir fijero de datos ---

open(daten_workspace):

--- determinar el vector a ----

ang_phi := args[1]:

ang_theta := args[2]:

phi := evalf(ang_phi):

theta :=evalf(ang_theta):

alpha2 := -1:

alpha5 := -1:

alpha6 := -1:

offset2 := -0.1396:

offset3 := Pi/6:

offset4 := 0:

offset5 := -Pi/2:

offset6 := Pi/2:

TCPoffset := 18:

a2 := 54:a3 := 32.4:a4 := 12:d5 := 4.8:

d6 := 21.2+TCPoffset:

xmin := 0: xmax := 130:

ymin := -130: ymax := 130:

```

zmin := 0: zmax := 0:

delta :=10:
EPS := 0.0001:

## ----- limites de las articulaciones -----
t1min := -Pi/2 : t1max:= Pi/2 :
t2min := alpha2*0 + offset2 : t2max:= alpha2*2*Pi/3 + offset2:
t3min := 0 + offset3 : t3max:= 1.92 + offset3:
t4min := -0.873 + offset4 : t4max:= 0.873 + offset4:
t5min := alpha5*(-0.917) + offset5 : t5max:= alpha5*0.917 + offset5:
t6min := alpha6*(-Pi) + offset6 : t6max:= alpha6*Pi + offset6:

## ----- parámetros de orientacion -----
ax := cos(phi)*cos(theta): ay := sin(phi)*cos(theta):
az :=-sin(theta):

for z from zmin by delta to zmax do
    i := 0:
    for x from xmin by delta to xmax do
        j := i: i := j + 1:
        for y from ymin by delta to ymax do
            ## -----
            ## ----- DHcin = DHreal * inv(Trans_d6) -----
            px := -ax*d6 + x:
            py := -ay*d6 + y:
            pz := -az*d6 + z:

            ## ----- Teta 1 -----
            if (px >= 0) then
                t1 := arctan(py,px):
            else
                t1 := -arctan(py,-px):

```

```

fi:

if (evalf(t1) > evalf(t1max)) then next
elif (evalf(t1) < evalf(t1min)) then next
fi:

f11a := evalf(cos(t1)*ax + sin(t1)*ay):
f12a := -az:
f13a := evalf(-sin(t1)*ax + cos(t1)*ay):
f11p := evalf( cos(t1)*px + sin(t1)*py):
f12p := -pz:

## ----- Teta 234 -----
t234 := arctan(f12a,f11a):

u := evalf(a4*cos(t234) + d5*sin(t234)):
v := evalf(a4*sin(t234) - d5*cos(t234)):

## eliminacion de inexactitudes ##
w := evalf((evalf((f11p-u)^2) + evalf((f12p-v)^2) - evalf(a2^2) - evalf(a3^2))/evalf(2*a2*a3)):

if (w>1) then w := 1
elif (w<0) then next
fi:

## ----- Teta 3 -----
t3 := evalf(arctan(sqrt(1-w^2),w)):

if (evalf(abs(t3-evalf(offset3)))<0.00001) then t3 := evalf(offset3)
fi:

if (evalf(t3) > evalf(t3max)) then next
elif (evalf(t3) < evalf(t3min)) then next

```

```

fi:

k := evalf(a2*cos(t3) + a3):
l := evalf(a2*sin(t3)):

e := evalf(2*I*f12p - 2*k*f11p):
f := evalf(-2*I*f11p - 2*k*f12p):
d := evalf(a4^2 + d5^2 - (f11p^2 + f12p^2 + k^2 + l^2)):

## ----- Teta 23 -----
t23_1 := -arctan(e,f) + arctan(d,-sqrt(e^2 + f^2 - d^2)):
t23_2 := arctan(e,-f) - arctan(d,-sqrt(e^2 + f^2 - d^2)):

t_val := cos(t23_1-t3)*a2+a4*cos(t234)+d5*sin(t234)+a3*cos(t23_1);

if ((t_val<(f11p+EPS)) and (t_val>(f11p-EPS))) then t23 := t23_1
else t23 := t23_2
fi:

## ----- Teta 4 -----
t4 := evalf(t234 - t23):

if (evalf(t4) > evalf(t4max)) then next
elif (evalf(t4) < evalf(t4min)) then next
fi:

## ----- Teta 2 -----
t2 := evalf(t23 - t3):

if (evalf(t2) < evalf(t2max)) then next
elif (evalf(t2) > evalf(t2min)) then next
fi:

```

```

f33a := f13a:
f32a := evalf(-sin(t23)*f11a + cos(t23)*f12a):
f31a := evalf(cos(t23)* f11a + sin(t23)*f12a):
f42a := f33a:
f41a := evalf(cos(t4)*f31a + sin(t4)*f32a):

## ----- Teta 5 -----
t5 := arctan(-f41a,f42a):

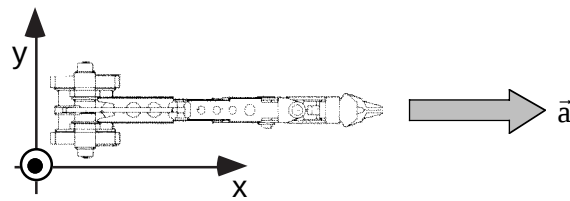
if (evalf(t5) < evalf(t5max)) then next
elif (evalf(t5) > evalf(t5min)) then next
fi:
## ----- puntos en fijero -----
writeln(x,y,z);

## -----
od: ## ende for y
od: ## ende for x
od: ## ende for z

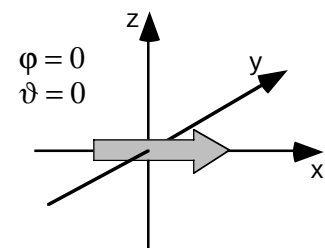
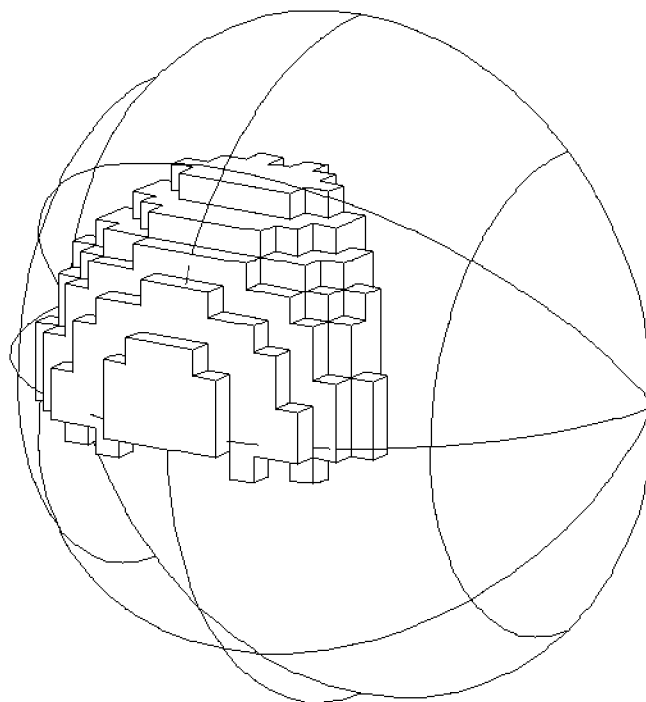
close(daten_workspace);
## ----- FIN -----
end;
```

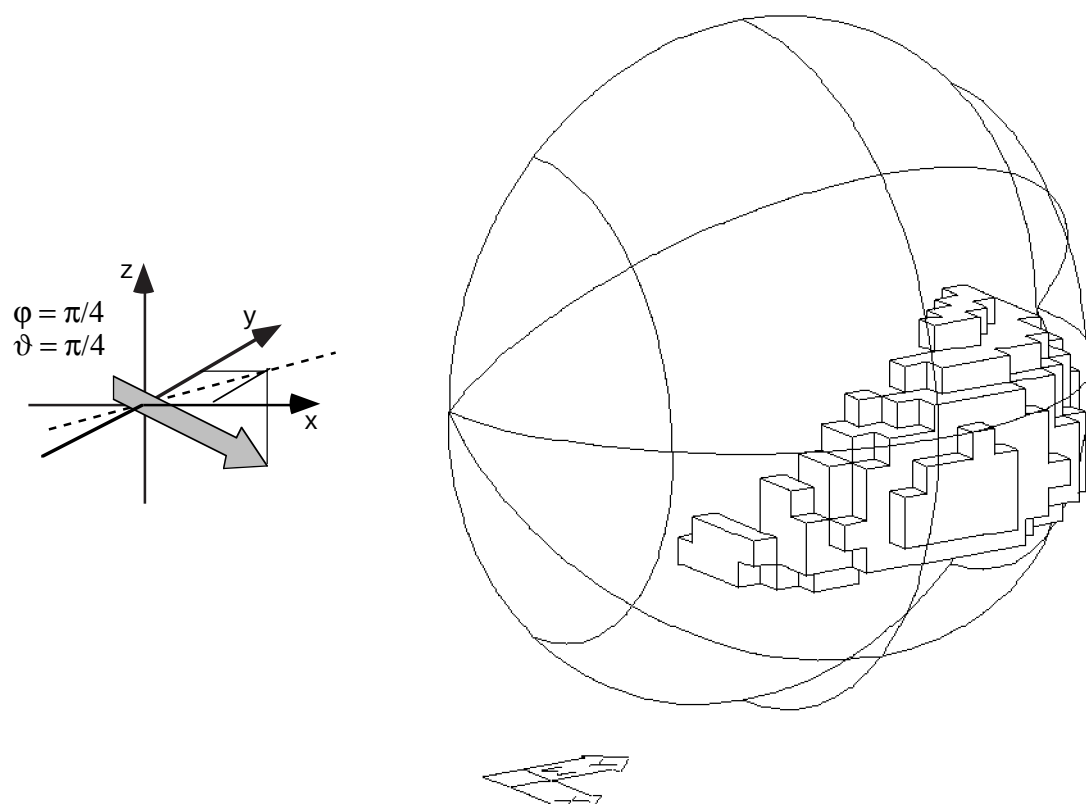
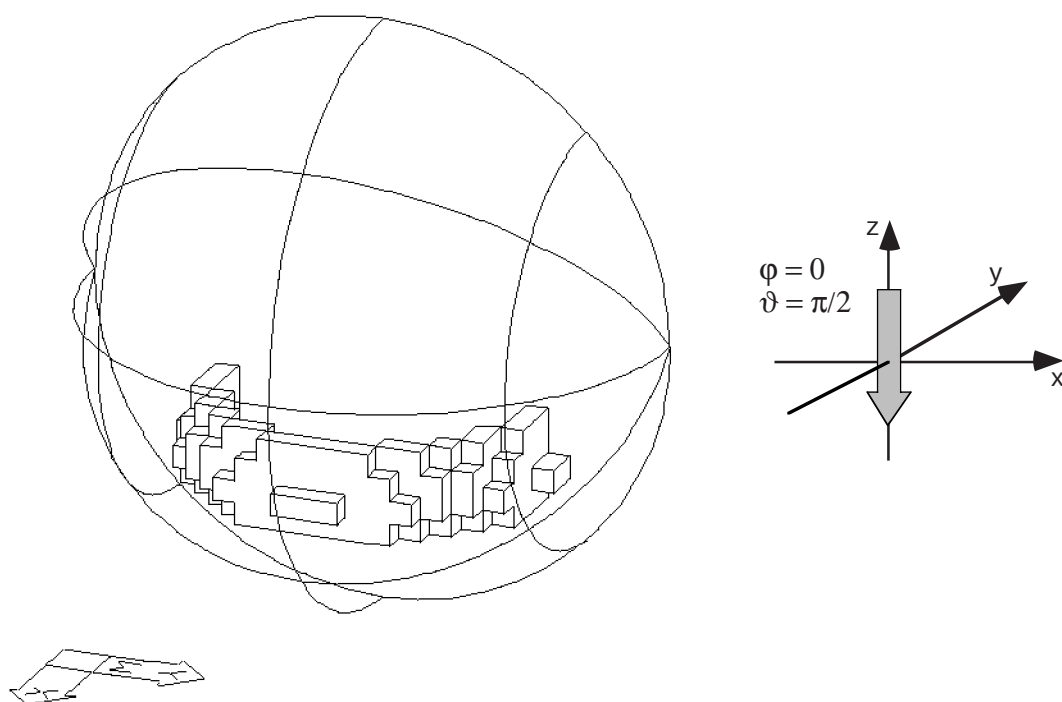
B.8. Visualización de espacios de trabajo para varias orientaciones

La siguiente figura muestra la orientación inicial, en la que basan los cálculos de los espacios de trabajo. La media esfera esbozada aproxima el espacio de trabajo sin tener en cuenta orientaciones.



orientación inicial del robot GRIPS





B.9. Representación del robot GRIPS en TOROS

B.10. Representación del sistema completo en TOROS

5. Bibliografía

- [1] Borrel, P., Leigois, A.
A Study of Manipulators Inverse Kinematic Solutions with Applications to Trayjectory Planning and Workspace Determination
IEEE '86, International Conference on Robotics and Automation, pages 1180-1185
San Fransisco, C.A. April 1986
- [2] Paul, P.
Robot Manipulators: Mathematics, Programming and Control
MIT Press, Cambrige, Mass. 1981
- [3] Alameldin, T. K.
Threedimensional Workspace Determination for redundant articulated Chains
Doctoral thesis, University of Pennsylvania, 1991
- [4] Dillmann, R., Huck, M.
Informationsverarbeitung in der Robotik
Springer, Berlin 1991
- [5] Becquet, M.
Computing of Robot's Workspace
ICAR '85, 1985, pages 289-295
- [6] Paul, R.P., Shimano, B., Mayer, G.E.
Kinematic Control Equations for Simple Maniplators
IEEE Transactions on Systems, Man and Cybernetics, 6. June 1981
- [7] Bronstein, Semendjajew
Taschenbuch der Mathematik
Teubner, Leipzig, 1979

[8] Kraft Telerobotics Inc.
11667 West 90th Street, Overland Park, KS 66214, USA
Description of GRIPS Force Feedback Manipulator System

[9] Balaguer, C., Rodríguez, F.
TOROS: Graphical Toolbox for Robot Simulation
International Workshop on Graphics & Robotics
Schloß Dagstuhl, Germany, 19.-22. April 1993

[10] Char, B.W., Geddes, K.O.,
MAPLE V Language Reference Manual
Springer Verlag, New York, 1991