

FABRIK

- Es un nuevo método heurístico que utiliza las posiciones previamente calculadas de las articulaciones para buscar las actualizaciones en un modo iterativo hacia adelante y hacia atrás.
- FABRIK consiste en minimizar el error del sistema ajustando cada ángulo común a la vez.
- El método propuesto se inicia a partir de la última articulación de la cadena y trabaja hacia delante, ajustando cada articulación a lo largo del camino.
- Después de eso, trabaja hacia atrás de la misma manera, con el fin de completar una iteración completa.
- Este método, en lugar de usar rotaciones de ángulo, trata de encontrar los lugares comunes como un problema de encontrar un punto en una línea; por lo tanto, se pueden ahorrar tiempo y cómputo.

Una iteración completa del algoritmo FABRIK

Input: The joint positions \mathbf{p}_i for $i = 1, \dots, n$, the target position \mathbf{t} and the distances between each joint $d_i = |\mathbf{p}_{i+1} - \mathbf{p}_i|$ for $i = 1, \dots, n-1$

Output: The new joint positions \mathbf{p}_i for $i = 1, \dots, n$.

1.1 % The distance between root and target

1.2 $dist = |\mathbf{p}_1 - \mathbf{t}|$

1.3 % Check whether the target is within reach

1.4 if $dist > d_1 + d_2 + \dots + d_{n-1}$ then

1.5 % The target is unreachable

1.6 for $i = 1, \dots, n-1$ do

1.7 % Find the nearest point on sphere, with centre the joint position \mathbf{p}_i and radius the distance d_i , from a point is space, \mathbf{t}

1.8 $\mathbf{p}_{i+1} = \text{NearestPointSphere}(\mathbf{p}_i, d_i, \mathbf{t});$

1.9 end

1.10 else

1.11 % The target is reachable; thus, set as \mathbf{b} the initial position of the joint \mathbf{p}_1

1.12 $\mathbf{b} = \mathbf{p}_1$

1.13 % Check whether the distance between the end effector \mathbf{p}_n and the target \mathbf{t} is greater than a tolerance.

$diff = |\mathbf{p}_n - \mathbf{t}|$

while $diff > tol$ do

1.16 % STAGE 1: FORWARD REACHING

1.17 % Set the end effector \mathbf{p}_n as target \mathbf{t}

$\mathbf{p}_n = \mathbf{t}$

1.19 for $i = n-1, \dots, 1$ do

1.20 % Find the nearest point on sphere, with centre the joint position \mathbf{p}_{i+1} and

radius the distance d_i , from a point is space, \mathbf{p}_i

1.21 $\mathbf{p}_i = \text{NearestPointSphere}(\mathbf{p}_{i+1}, d_i, \mathbf{p}_i);$

```

1.22      end
1.23      % STAGE 2: BACKWARD REACHING
1.24      % Set the root p1 its initial position.
1.25      p1 = b
1.26      for i = 1, . . . , n-1 do
1.27          % Find the nearest point on sphere, with centre the joint position pi and
          radius the distance di , from a point is space, pi+1
          pi 1.28 = NearestPointSphere(pi , di , pi+1);
1.29      end
1.30      dif A = |pn 1.30 - t|
1.31      end
1.32 end
1.33

1.34 % Where the function NearestPointSphere(X,Y,Z), finds the nearest point on a sphere
from a point in space. X is the sphere's centre, Y is the sphere's radii and Z is the point
in space.

```

Asumimos que p_1, \dots, p_n son las posiciones conjuntas de un manipulador.

Para el caso simple donde sólo existe un único efecto final, tome p_1 como la articulación de la raíz y p_n como el efecto final.

El objetivo es t , y la posición de base inicial es b .

En primer lugar calcular las distancias entre cada articulación

$$d_i = |p_{i+1} - p_i| \text{ for } i = 1, \dots, n - 1.$$

Entonces, para comprobar si el objetivo es alcanzable o no, encontramos la distancia entre la raíz y el objetivo, $dist$, y si esta distancia es menor que la suma total de todas las distancias entre conjuntos, el objetivo está a nuestro alcance; de lo contrario, es inalcanzable.

Si el objetivo está dentro de alcance, una iteración completa está constituida por dos etapas. En la primera etapa, el algoritmo estima cada posición de la articulación a partir del efecto final, p_n , moviéndose hacia el interior, a la base del manipulador, p_1 . Por lo tanto, dejamos que la nueva posición del efecto final sea la posición de destino, $p'_n = t$.

La nueva posición conjunta de la $(n-1)$ -ésimo, p'_{n-1} , se asigna como el punto de la esfera Σ_{n-1} más cercano, con el centro de la posición conjunta y radios p'_{n-1} , la distancia d_{n-1} de la posición conjunta p_{n-1} .

Del mismo modo, la nueva posición conjunta de la $(n - 2)$ -ésimo, p'_{n-2} , se selecciona como el punto sobre la esfera Σ_{n-2} más cercana, con el centro de la articulación de la posición p'_{n-1} y radios de la distancia d_{n-2} de la articulación p_{n-2} .

El algoritmo continúa hasta que se calculan todas las nuevas posiciones comunes, incluyendo la raíz, p'_1 . El punto en una esfera de un punto en el espacio más cercano se encuentra simplemente tomando un punto a lo largo de la línea que une el centro de la esfera hasta el punto, que tiene distancia desde el centro igual al radio de la esfera.

Una iteración completa finaliza cuando se repite el mismo procedimiento, pero esta vez a partir de la articulación de la raíz y en movimiento hacia el exterior hasta el efecto terminal. Por lo tanto, dejamos que la nueva posición de la primera articulación, p''_1 , sea su posición inicial b . Entonces, la nueva posición conjunta p''_2 se asigna como el punto más cercano en la esfera Σ_1 , con el centro de p''_2 y radios la distancia conjunta d_1 de la p'_2 . Este

procedimiento se repite para todas las articulaciones restantes, incluyendo el efecto de extremo.

Algoritmo paso a paso:

1. Comprobar si el objetivo es alcanzable; Para ello se suman las distancias que hay entre cada articulación. Si esta distancia es mayor o igual que la distancia que hay entre la primera articulación y el objetivo quiere decir que el objetivo es alcanzable.

Si el objetivo está dentro de alcance, una iteración completa está constituida por dos etapas:

2. En la primera etapa, el algoritmo estima cada posición de la articulación a partir del efecto final:
 - 2.1 Primero dejamos el efecto final se coloque en la posición del objetivo.
 - 2.2 Se traza una línea que pasa a través de la articulación anterior y la del efecto final.
 - 2.3 Se coloca la articulación anterior sobre la línea trazada anteriormente a la distancia de su link.
 - 2.4 Se repite el proceso hasta que se han colocado todas las articulaciones.

3. La articulación base no debe ser diferente de su posición inicial.
Por tanto, en la segunda etapa el algoritmo recoloca la articulación base a su origen recalculando las posiciones de las demás articulaciones.

El procedimiento es exactamente el mismo, pero en sentido contrario.

- 3.1 Se coloca la articulación base en su posición inicial.
- 3.2 Se traza una línea que pasa a través de la articulación base y la siguiente articulación.
- 3.3 Se coloca la articulación siguiente sobre la línea trazada a la distancia de su link.
- 3.4 Se repite el proceso hasta que se han recolocado todas las articulaciones.

Esto completa una sola iteración. Se puede apreciar que el efecto final se habrá desplazado del objetivo. Para corregir esto se ejecuta el algoritmo desde el paso 2 hasta que el efecto final queda sobre dicho objetivo o cerca de este con un mínimo fallo.

