



## MODELACIÓN Y CONTROL EN ESPACIO DE ESTADO-DE UN SERVOMECANISMO USANDO MATLAB

**Wilmer Sunz (1); Jorge Castellanos (2); César Peraza (3); Francisco Arteaga (4)**

**(1) Universidad de Carabobo, Escuela de Ing. Eléctrica, Unidad de Automatización Indust. ; Email: wilmsanz@uc.edu.ve (2) Universidad de Carabobo, Facultad de Ciencias y Tecnología, Dpto. de Computación. Email: jcasteld@uc.edu.ve (3) Instituto Universitario de Tecnología Valencia. Dpto. de Electricidad Email: cperaza@postgrado.uc.edu.ve (4) Universidad de Carabobo, Escuela de Ing. Eléctrica, Unidad de Automatización Industrial. Email: farteaga@uc.edu.ve**

### Resumen

Este trabajo presenta el estudio de un servomecanismo desarrollado sobre la base de un motor dc, controlado por campo o por armadura. Se realiza el análisis y control del sistema modelado en espacio de estado y se obtiene la respuesta del sistema mediante los diseños con realimentación de los estados y de sistema de seguimiento como observadores de estado.

**Palabras claves:** servomecanismos, simulación en Matlab, realimentación de estado, diseño de observador.

### Abstract

This works presents a servomechanism study developed on a dc motor basis, controlled by field or armature. The system analysis and control in state space model is performed and system responses with state feedback design and tracking systems with state observers are obtained.

**Key words:** servomechanisms, Matlab simulation, state feedback, observer design.

## 1. INTRODUCCIÓN

Los sistemas tipo servo, mejor conocidos como servomecanismos, son aquellos que implementan una acción de seguimiento de la variable de salida respecto a una referencia. En ellos, los actuadores, (elementos que llevan a cabo la acción) son motores de corriente directa (dc) conocidos como servomotores.

Entre las características de los servomotores puede destacarse una inercia pequeña o bien una alta relación entre el par y la inercia, por lo cual resultan útiles en aplicaciones tan variadas como las correspondientes al campo de la robótica o las que involucran labores de fresado y mecanizado de piezas

El manejo de tal elemento actuador corre por cuenta de un servo manejador, que consiste en un subsistema electrónico controlador de movimiento. Dicho sistema debe admitir la programación del posicionamiento por puntos y de otras variables derivadas como aceleración y velocidad.

Delineadas así la importancia y multiplicidad de aplicaciones de los servomecanismos, se comprenderá que muchos folios difícilmente proporcionarían una explicación cabal acerca de los mismos. Huelga entonces subrayar la pertinencia de un artículo como este, donde se ofrece una modelación en espacio de estados y un análisis de la respuesta temporal de los servomecanismos, ante estímulos como el escalón.

## 2. MODELACIÓN DE LOS PARÁMETROS ELÉCTRICOS: DE UN MOTOR DC.

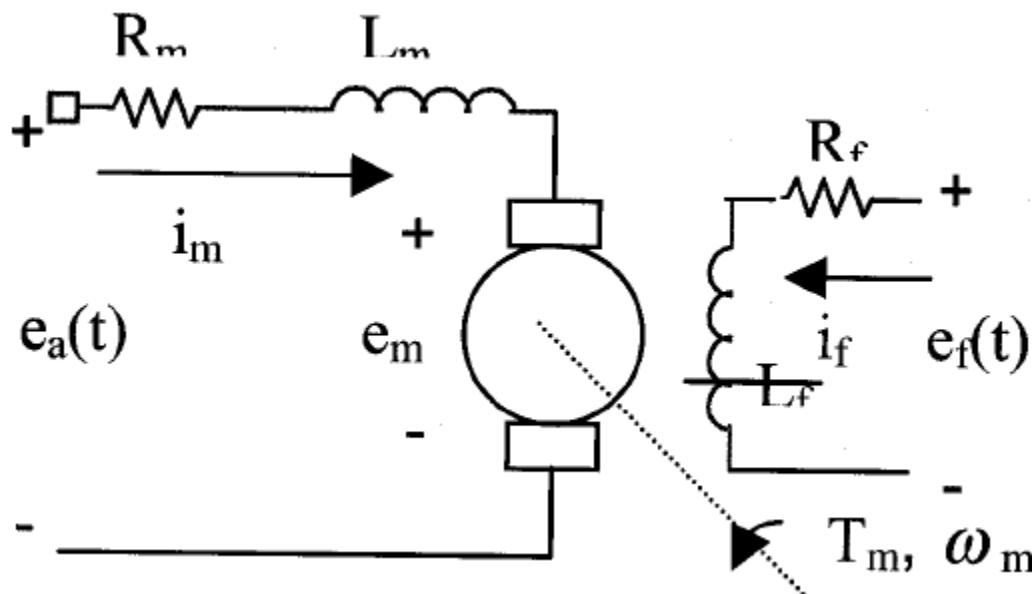
Las ecuaciones que describen el comportamiento de motor DC son relativas al par en el eje ( $T_m$ ) y a la fuerza contra electromotriz que se genera en sus bornes como reacción a la excitación en su armadura:

$$T_m(t) = K_1 \Phi(t) \cdot i_m(t) \quad (1)$$

$$e_m(t) = K_2 \Phi(t) \cdot \omega_m(t) \quad (2)$$

Las variables involucradas son las que se indican en la Figura 1,  $K_1$ ,  $K_2$  son constantes y  $F(t)$  es el flujo magnético en el campo de excitación, dependiente de la tensión de campo  $e_f(t)$ .

El campo magnético de excitación dc puede estar alimentado por la corriente que circula en la armadura, puede tener alimentación separada o bien puede provenir de un imán permanente. La ilustración de la Figura 1 corresponde al caso más general: excitación separada y variable en el tiempo.



**Fig.1 Esquema de un Servomotor con excitación separada.**

Si la excitación del campo es constante ( $e_f$  igual a una tensión dc) o proviene de imanes permanentes, el flujo magnético  $F$ , también será constante; por lo que el funcionamiento del motor depende exclusivamente de la corriente que circula por su armadura. En este caso se implementa un Control de armadura.

Si, al contrario, se mantiene constante la corriente de armadura, el control puede lograrse manipulando el campo magnético de excitación, a través de la variable  $e_f(t)$ . Se habla entonces de un servomotor Controlado por campo.

El control de armadura es mucho más usado que el control por campo. Esto se debe a la dificultad que implica implementar una fuente de corriente constante, contra la facilidad relativa de hacer lo propio para una fuente de tensión constante. Sin embargo, el control de campo encuentra aplicaciones prácticas en algunos sistemas donde se busca regular la velocidad de giro en el eje del motor. Por esa razón se procede a continuación con el estudio de ambos.

### 3. MODELACIÓN DE LOS PARÁMETROS MECÁNICOS DEL MOVIMIENTO ROTACIONAL.

La Figura 2 muestra a un motor dc con su eje acoplado a una carga mecánica. En un sistema como este, el giro del motor depende de parámetros como la inercia de la carga y la resistencia proveniente de amortiguadores y elementos rígidos.

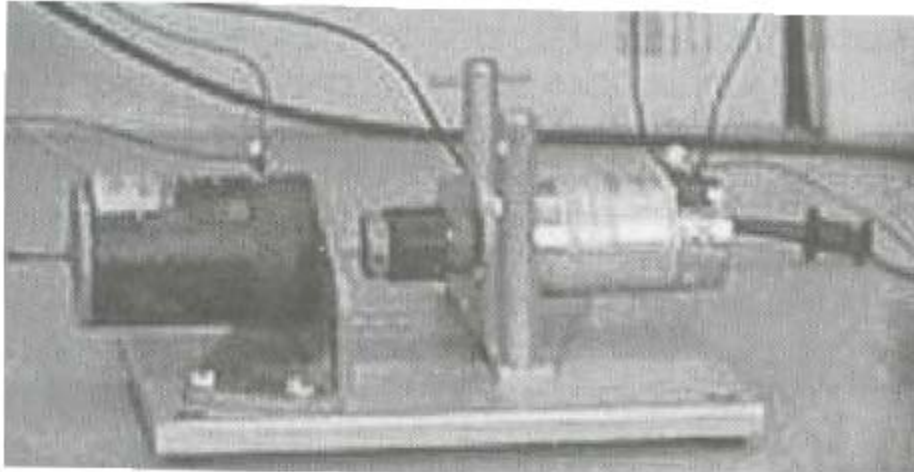


Fig. 2 Acoplamiento de un servomotor con una carga mecánica

#### El momento de inercia.

La figura 3 muestra la aplicación de un Torque sobre un cuerpo que posee un momento de inercia  $J$ , a través de una barra que representa el eje de un servomotor.



El resultado es un desequilibrio que se manifiesta como un cambio en la posición, la velocidad y la aceleración, angulares ( $\theta$ ,  $\omega$  y  $a$ ).

La sumatoria de tórces determina que el único estímulo existente ( $T_m$ ) es directamente proporcional a la aceleración angular del eje:

$$T_m = T_j = J \frac{d^2 \theta_m}{dt^2} \quad (3)$$

### Resistencia por rigidez.

La figura 4 ilustra el caso extremo de una barra empotrada sobre la que se aplica un torque externo. La propiedad de la barra, de recuperar su posición angular original, luego de la torsión que se le causa, se define como un torque de restitución, este torque es directamente proporcional al ángulo girado [1].

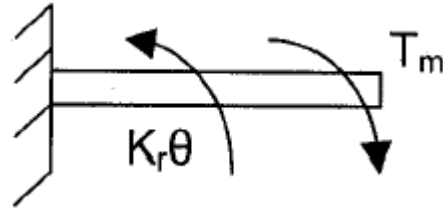


Fig.4 Torque resistivo por rigidez

Si  $K_r$  es la constante de restitución del sistema, se tiene que

$$T_m - K_r\theta_m = J \frac{d^2\theta_m}{dt^2} \quad (4)$$

### Resistencia por amortiguamiento.

En la figura 5 se nota la colocación de un elemento acoplado al eje del motor, que desliza sobre un líquido viscoso, lo que determina la existencia de un torque resistivo que es proporcional a la velocidad angular con la cual se efectúa el movimiento del eje.

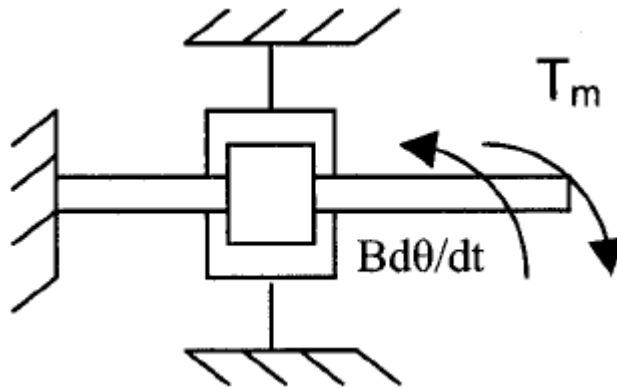


Fig.5 Torque resistivo por amortiguamiento

$$T_m - B \frac{d\theta_m}{dt} = J \frac{d^2\theta_m}{dt^2} \quad (5)$$

En la ecuación anterior (5),  $B$  es la constante de proporcionalidad entre la velocidad angular del eje y la resistencia que ofrece la sustancia viscosa sobre la que este se desliza.

#### 4.MODELACIÓN DE UN SERVOMECHANISMO CONTROLADO POR ARMADURA.

El sistema completo de un servomotor controlado por armadura se muestra en la figura 6.

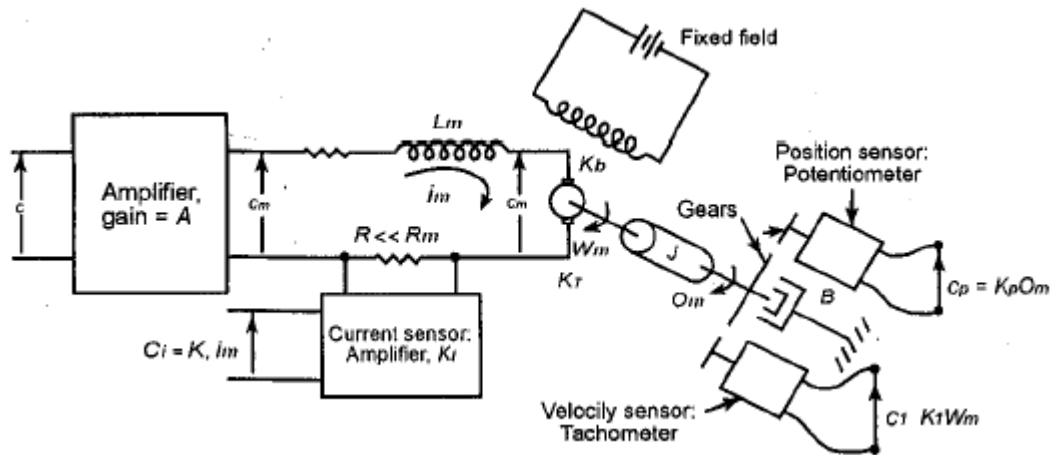


Fig. 6 Servomotor controlado por armadura

El control de armadura se implementa manteniendo constante la excitación del campo y manipulando la tensión de armadura  $e_a$  que se encuentra a la entrada del circuito.

Es decir:

$$\Phi(t) = K_f \cdot i_f(t) = \text{cte.}$$

$$T_m(t) = K_t \cdot i_m(t) \quad (6)$$

$$e_m(t) = K_e \cdot \omega_m(t) \quad (7)$$

Del circuito de armadura mostrado en la Figura 1:

$$L_m \frac{di_m}{dt} + R_m i_m + e_m = e_a \quad (8)$$

En el sistema rotacional que existe sobre el eje del motor, se obtiene la ecuación:

$$T_m = J \frac{d^2 \theta_m}{dt^2} + B \frac{d\theta_m}{dt} + K_r \theta_m \quad (9)$$

A través del método de la representación en variables de estado, se deben definir las ecuaciones que relacionan sus entradas y salidas; las cuales se identificarán con la tensión aplicada a la armadura ( $e_a$ ) y la posición angular del eje del motor ( $\theta_m$ ), respectivamente. Sobre esta base se procede a manipular las ecuaciones planteadas anteriormente, quedando como resultado:

$$\omega = \frac{d\theta}{dt} \quad (10)$$

$$\alpha = -\frac{K_r}{J}\theta - \frac{B}{J}\omega + \frac{K_t}{J}i_m \quad (11)$$

$$\frac{di_m}{dt} = -\frac{K_e}{L_m}\omega - \frac{R_m}{L_m}i_m + \frac{1}{L_m}e_a \quad (12)$$

Si se seleccionan como variables de estado a la posición angular ( $\theta$ ), la velocidad ( $\omega$ ) y la corriente de armadura ( $i_m$ ), se tiene:

$$\begin{array}{l} x_1 = \theta \Rightarrow \dot{x}_1 = \omega \\ x_2 = \omega \Rightarrow \dot{x}_2 = \alpha \\ x_3 = i_m \Rightarrow \dot{x}_3 = \frac{di_m}{dt} \\ u = e_a \\ y = \theta \Rightarrow y = x_1 \end{array} \left| \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{K_r}{J}x_1 - \frac{B}{J}x_2 + \frac{K_t}{J}x_3 \\ \dot{x}_3 = -\frac{K_e}{L_m}x_2 - \frac{R_m}{L_m}x_3 + \frac{1}{L_m}u \\ y = x_1 \end{array} \right.$$

La representación en espacio de estados queda así:

$$\dot{\underline{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{K_r}{J} & -\frac{B}{J} & \frac{K_t}{J} \\ 0 & -\frac{K_e}{L_m} & -\frac{R_m}{L_m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L_m} \end{bmatrix} \underline{u}$$

$$\underline{y} = [1 \quad 0 \quad 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Para efectuar el análisis de este sistema se considerarán los aspectos siguientes:

- ⇒ Función de transferencia.
- ⇒ Estudio de la respuesta del sistema, al escalón, sin realimentación de estados.
- ⇒ Observabilidad.
- ⇒ Controlabilidad.
- ⇒ Diseño de un sistema de control mediante asignación de polos.
- ⇒ Estudio de la respuesta al escalón con Realimentación Lineal de Estados (RLE).
- ⇒ Introducción de entrada de referencia e implementación de Control de Seguimiento (Tracking Control).
- ⇒ Diseño de observadores de estados.

Como herramienta fundamental para llevar a cabo el análisis se usará el software Matlab. Los comandos que se mostrarán corresponden a un m file (extensión .m), creado por los autores (sobre la base de otros desarrollados en diversas publicaciones [2],[3],[4] y [5]) para efectuar un diseño de control por armadura de un servomotor mediante la introducción de los parámetros que definan al sistema.

#### 4.1 Construcción de la función de transferencia y obtención de los polos del sistema..

La función de transferencia del sistema modelación en espacio de estados se obtiene mediante el comando ss2tf. Su sintaxis es:

$$[\text{num}, \text{den}] = \text{ss2tf}(A, B, C, D)$$

Este comando convierte el sistema de Espacio de Estados:

$$\begin{aligned}\underline{\dot{x}} &= A\underline{x} + B\underline{u} \\ \underline{y} &= C\underline{x} + D\underline{u}\end{aligned}$$

a la función de transferencia:

$$\frac{Y(s)}{U(s)} = \text{i-ésimo elemento de } [C(sI - A)^{-1} B + D]$$

Los polos del sistema se obtienen determinando las raíces del polinomio característico de la función de transferencia  $G(s)=Y(s)/U(s)$ . La expresión a evaluar es:

$$|sI - A| = 0$$

El comando de Matlab correspondiente es eig(A), pero también puede emplearse roots(den), donde den es un vector que contiene los coeficientes del polinomio del denominador de G(s). Debe observarse que den se obtuvo al usar el comando ss2tf.

#### 4.2. Estudio de la respuesta del sistema, al escalón, sin realimentación de estados.

Para graficar la respuesta al escalón, debe definirse previamente un vector  $t$ , que contenga los valores en los cuales se evaluarán las variables de estado del sistema. Este vector inicialmente será igual a cero, tendrá un valor final suministrado por el usuario y un incremento, también definido por el usuario:

$$t=[0.0:\text{incremento}:\text{valor final}].$$

La entrada escalón estará conformada por un vector cuyas componentes son todas iguales a 1; y tendrá tantas componentes como valores en el tiempo tenga el vector  $t$ .

$$u=\text{ones}(1,\text{length}(t))$$

Como las condiciones iniciales tendrán valores suministrados por el usuario se formará un vector:

$$x0=[ci1; ci2; ci3]$$

Para simular la salida del sistema se hace uso del comando de Matlab Isim y se asigna el resultado a una matriz:

$$y=\text{lsim}(A,B,C,D,u,t,x0)$$

Finalmente, la gráfica esperada se obtiene al utilizar el comando plot, del siguiente modo:

$$\text{plot}(t,y)$$

La siguiente secuencia ilustra el uso de todos los comandos explicados.

```
[num,den] = ss2tf(A,B,C,D);
polos=eig(A);
% también es válido roots(den)
t=[0.0:incremento:valor final];
u=ones(1,length(t));
x0=[ci1; ci2; ci3];
y=lsim(A,B,C,D,u,t,x0);
plot(t,y);
```

#### 4.3. Estudio de la Controlabilidad y la Observabilidad.

La Controlabilidad de un sistema puede entenderse como la propiedad de obtener cambios deseados en sus variables de estado a través de la manipulación de sus entradas

La condición para la Controlabilidad completa de sus variables de estado, es que la matriz:



$$M = [B \ AB \ A^2B \ \dots \ A^{n-1}B] \quad (13)$$

sea de rango  $n$ , donde  $n$  es el número de filas o columnas de la matriz  $A$ . Esto significa que  $M$  debe tener  $n$  vectores columna, linealmente independientes.

Un modo de verificar esta condición con Matlab es a través del uso de los comandos `ctrb` y `rank`.

$$M = \text{ctrb}(A, B)$$

$$\text{rank}(M)$$

Con la primera de estas dos líneas se obtiene la matriz  $M$ , y con la segunda, su rango.

Por su parte, la Observabilidad puede identificarse como la propiedad que tiene un sistema para lograr la influencia directa o indirecta de sus salidas mediante la manipulación de sus variables de estado.

La condición de completa Observabilidad de un sistema se verifica si puede construirse una matriz:

$$N = [C^T \ A^T C^T \ (A^T)^2 C^T \ \dots \ (A^T)^{n-1} C^T] \quad (14)$$

Tal que el rango de  $N$  sea igual a  $n$ .

Usando Matlab es muy simple determinar a la matriz  $N$  y su rango, mediante los comandos:

$$N = \text{obsv}(A, C)$$

$$\text{rank}(N)$$

#### 4.4. Diseño de un sistema de control regulatorio mediante asignación de polos.

El diseño de un sistema de control mediante la asignación de polos busca lograr la compensación de perturbancias para devolverlo a una condición inicial de equilibrio (Regulación). Este procedimiento es conocido como Realimentación Lineal de Estados (RLE) y consiste en aplicar una transformación que permita ubicar los polos del sistema donde se deseen.

Esto se logra mediante la realimentación de las variables de estado, a través de una matriz de ganancia ( $K$ ), para redefinir las entradas del sistema, ahora en lazo cerrado.

La figura 7 sirve como ilustración del concepto.

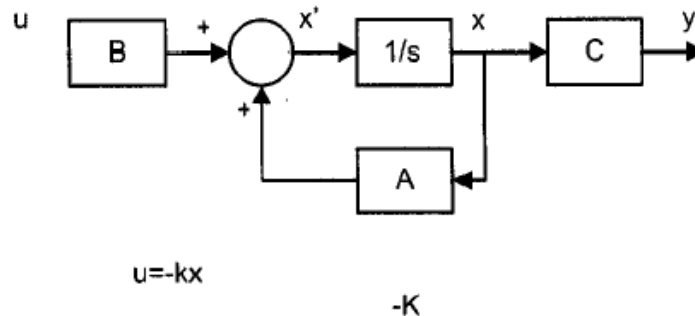


Figura 7: Sistema de control en lazo cerrado con  $u = -Kx$

La realimentación requerida puede implementarse al acoplar un potenciómetro en el eje del motor, de manera que las variaciones de la posición angular se traduzcan en cambios de tensión. Para la velocidad se usa un tacómetro y la corriente de armadura puede sensarse indirectamente a través de la tensión caída en una pequeña resistencia (recuérdese que las tres variables de estado son  $\theta$ ,  $\omega$  e  $i_m$ ).

Una vía expedita para lograr el diseño deseado consiste en los siguientes pasos:

⇒ Verificar la controlabilidad del sistema

Obtener un vector formado por los coeficientes del polinomio característico de la matriz A ( $|sI-A| = s^n + a_1 s^{n-1} + a_2 s^{n-2} + \dots + a_{n-1} s + a_n$ ). Con Matlab, esto se logra haciendo uso del comando poly. Por ejemplo, en la expresión siguiente, se forma el vector cpca con estos coeficientes:

$$\text{cpca} = \text{poly}(A)$$

⇒ Determinar la matriz de transformación (T) que permite expresar la ecuación de estado en su Forma Canónica Controlable (FCC). T se obtiene como el producto de otras dos matrices: M y W. M es la matriz de controlabilidad y W tiene la forma que se muestra a continuación:

$$W = \begin{bmatrix} a_{n-1} & a_{n-2} & \dots & a_1 & 1 \\ a_{n-2} & a_{n-3} & \dots & 1 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ a_1 & 1 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \end{bmatrix}$$

⇒ Definir los polos deseados para el sistema en lazo cerrado y formar con ellos un polinomio característico. Previamente debe definirse una matriz (J) que tenga en la diagonal los valores de los polos deseados.

$$J = \begin{bmatrix} \mu_1 & 0 & 0 \\ 0 & \mu_2 & 0 \\ 0 & 0 & \mu_3 \\ \vdots & \vdots & \vdots \end{bmatrix}$$

En la próxima expresión, los polos deseados se representan como  $\mu_i$  y los coeficientes del polinomio característico resultante, mediante  $a_i$ .

$$(s-\mu_1)(s-\mu_2)\dots(s-\mu_n) = s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n \quad (15)$$

El vector con los coeficientes  $a_i$  (cpca) se forma con el comando poly.

$$\text{cpda}=\text{poly}(\mathbf{J})$$

⇒ La matriz de ganancias de realimentación se obtiene a partir de la Ecuación 16.

$$\mathbf{K}=[(\alpha_n-a_n) \ (\alpha_{n-1}-a_{n-1}) \ \dots \ (\alpha_1-a_1)]*\mathbf{T}^{-1} \quad (16)$$

En resumen, para un sistema con  $n = 3$ ; el grupo de comandos de Matlab que permiten seguir los pasos descritos para lograr el diseño de una matriz de realimentación de estado que garantice la ubicación de los polos del sistema donde se desee, son:

```

cpca = poly(A);
a1 =cpca(2); a2=cpca(3); a3 =cpca(4);
W=[a2 a1 1;a1 1 0; 1 0 0 ];
T =M*W;
J =[mu1 0 0; 0 mu2 0;0 0 mu3 ];
cpda=poly(J);
alfa1 =cpda(2); alfa2=cpda(3); alfa3 =cpda(4);
K=[alfa3-a3 alfa2-a2 alfa1-a1 ];
K1=K(1); K2 =K(2); K3 =K(3);

```

A continuación se muestra el uso de los comandos descritos en un ejemplo práctico. Se inicia tomando como base los parámetros necesarios para formar las matrices del sistema:

$$\begin{aligned}
R_m &= 0.2 \ \Omega \\
L_m &= 5 \ \text{mH} \\
K_e &= 0.055 \ \text{Vs/rad} \\
K_t &= 6 \times 10^{-5} \ \text{Nm/A} \\
K_r &= 0.01 \ \text{Nm/rad} \\
B &= 0.005 \ \text{Nms/rad} \\
J &= 0.0044 \ \text{Nms}^2/\text{rad}
\end{aligned}$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{K_r}{J} & -\frac{B}{J} & \frac{K_t}{J} \\ 0 & -\frac{K_e}{L_m} & -\frac{R_m}{L_m} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -2.273 & -9.091 & 0.014 \\ 0 & -11 & -40 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \frac{1}{L_m} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 200 \end{bmatrix} \quad C = [1 \quad 0 \quad 0] \quad D = \underline{0}$$

```

den =
    1.0000  41.1364  47.8773  90.9091
» g=tf(num,den)
Transfer function:
              1.847e-013 s + 2.727
              -----
              s^3 + 41.14 s^2 + 47.88 s + 90.91
» polos=eig(A)
polos =
    -0.5701 + 1.3957i
    -0.5701 - 1.3957i
    -39.9961
» t=[0.0:0.01:30];
» u=ones(1,length(t));
» x0=[0;0;0];
» y=lsim(A,B,C,D,u,t,x0);
» plot(t,y)

```

Respuesta del servomotor no realimentado, para entrada escalon

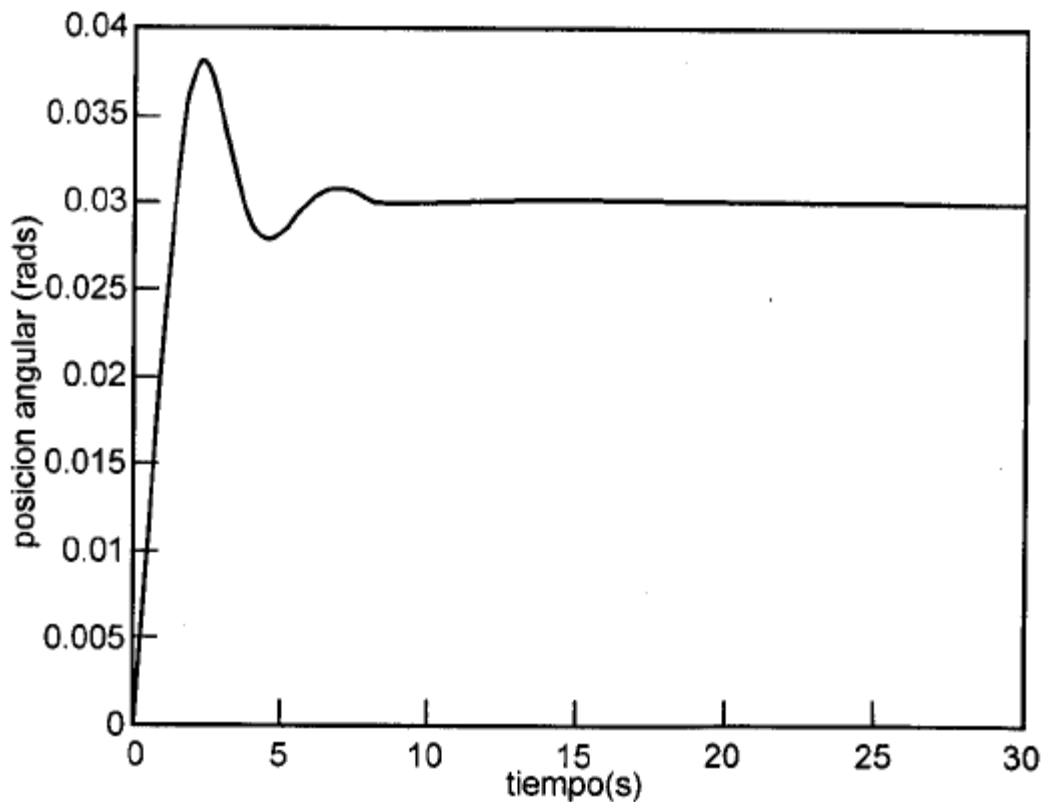


Figura 8: Respuesta temporal del servomotor controlado por armadura, no realimentado.

```

» M=ctrb(A,B)
M =
1.0e+005 *
    0    0  0.0000
    0  0.0000 -0.0011
    0.0020 -0.0800  3.1997
» rank(M)
ans =
    3
» % El sistema es de estado completamente controlable
» N=obsv(A,C)
N =
1.0000    0    0
    0  1.0000    0
   -2.2727 -1.1364  0.0136
» rank(N)
ans =
    3
» % El sistema es de estado completamente observable
» cpc = poly(A)
cpc =
    1.0000  41.1364  47.8773  90.9091
» a1 =cpc(2); a2=cpc(3); a3 =cpc(4);
» W =[a2 a1 1;a1 1 0; 1 0 0 ]

```

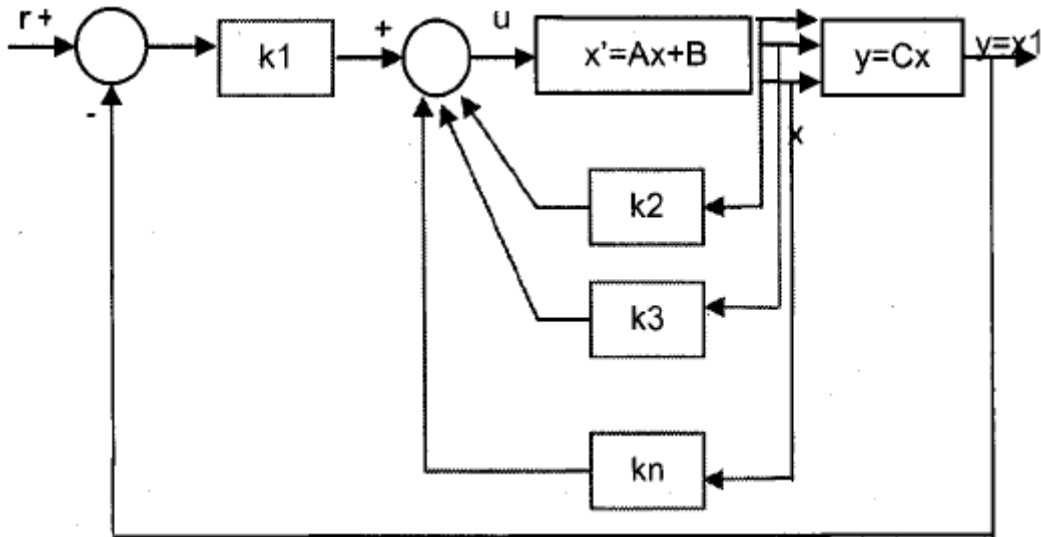
```

W =
  47.8773  41.1364  1.0000
  41.1364  1.0000   0
  1.0000   0   0
» T = M*W
T =
  2.7273   0   0
 -0.0000  2.7273   0
 454.5455 227.2727 200.0000
» J = [-2+3.4i 0 0; 0 -2-3.4i 0; 0 0 -10]
J =
 -2.0000 + 3.4000i   0   0
   0   -2.0000 - 3.4000i   0
   0   0   -10.0000
» % Los polos deseados se ubicarán en -2±3.4i, y -10
» cpda=poly(J)
cpda =
 1.0000 14.0000 55.5600 155.6000
» alfa1=cpda(2); alfa2=cpda(3); alfa3=cpda(4);
» alfa1
alfa1 =
 14
» alfa2
alfa2 =
 55.56
» alfa3
alfa3 =
 155.6
» Krle=[alfa3-a3 alfa2-a2 alfa1-a1]*T^-1;
Krle =
 -20.7384 14.1238 -0.1357
Krle(1)
ans =
 -20.7384
» Krle(2)
ans =
 14.1238
» Krle(3)
ans =
 -0.1357

```

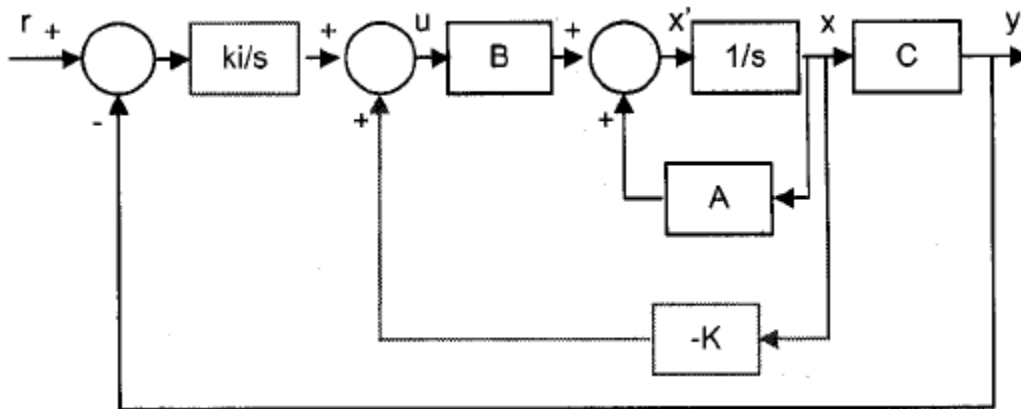
#### 4.5. Diseño del sistema de seguimiento mediante asignación de polos.

El procedimiento para diseñar un sistema de control de seguimiento es, esencialmente, el mismo empleado para RLE. La diferencia puede entenderse como una variante en la disposición de los elementos de la matriz de ganancias calculada, dentro del diagrama de bloques definitivo. La figura 9 ilustra esta situación.



**Figura 9: Sistema de seguimiento Tipo 1. Planta con integrador**

Los comandos de Matlab que conducen al cálculo de la matriz K ya se han descrito. Sin embargo, debe observarse que tal procedimiento es válido si el sistema original es tipo 1, es decir, si tiene un polo en el origen (un integrador). Puede demostrarse que esto se cumple para servomotores en los que puede despreciarse el torque por rigidez que se opone a la rotación del eje ( $K_r = 0$ ). En el caso más general, no ocurre así, siendo necesario proceder conforme lo ilustrado en la figura 10.



**Figura 10: Sistema de seguimiento Tipo 1. Planta sin integrador**

El diseño incluye la inserción del polo en el origen que convierte al sistema en tipo 1. Además, debe obtenerse una constante de integración ( $K_i$ ) paralelamente a la matriz de ganancias K. Esto requiere la introducción de un polo adicional ( $m_4$ ) que amplía el tamaño de la matriz J a  $4 \times 4$ . Se obtendrá entonces una matriz de ganancias ampliada:

$$\hat{K} = [K \mid K_i] \quad (17)$$

En la última expresión K es la matriz de ganancias del lazo de realimentación y  $K_i$  es la constante del integrador añadido.

A continuación se lista la secuencia de comandos de Matlab para calcular las constantes mencionadas.

```
A1=[A zeros(3,1);-C 0];
B1=[B;0];
Mservo=[B1 A1*B1 A1^2*B1 A1^3*B1];
rgo=rank(Mservo);
```

Se han formado las matrices A1 y B1, que corresponden al sistema ampliado, para verificar su controlabilidad tenemos la siguiente secuencia de comandos:

```
if rgo == 4,
J=[mu1 0 0 0;0 mu2 0 0;0 0 mu3 0; 0 0 0 mu4];
phi=polyvalm(poly(J),A1);
K=[0 0 0 1]*(inv(Mservo))*phi;
K1=K(1)
K2=K(2)
K3=K(3)
KI=-K(4)
KK=[K1 K2 K3];
```

Si se verifica que el nuevo sistema es completamente controlable se procede con el diseño propiamente dicho. En este caso se utiliza la fórmula de Akcermann, la cual requiere el cálculo de un polinomio matricial característico phi:

$$phi = [A1^4 + alfa1*A1^3 + alfa2*A1^2 + alfa3*A1 + alfa4*I] \quad (18)$$

En esta última expresión los coeficientes alfa son el resultado de poly(J). El comando polyvalm de Matlab facilita este cálculo

```
% Matrices del sistema en lazo cerrado
AA=[A-B*KK B*KI;-C 0]
BB=[0;0;0;1]
CC=[C 0]
DD=[0]
```

Las matrices AA, BB, CC y DD corresponden al sistema en lazo cerrado de la figura 11. Ellas son utilizadas en la secuencia de comandos que permiten graficar la respuesta transitoria para una entrada escalón.

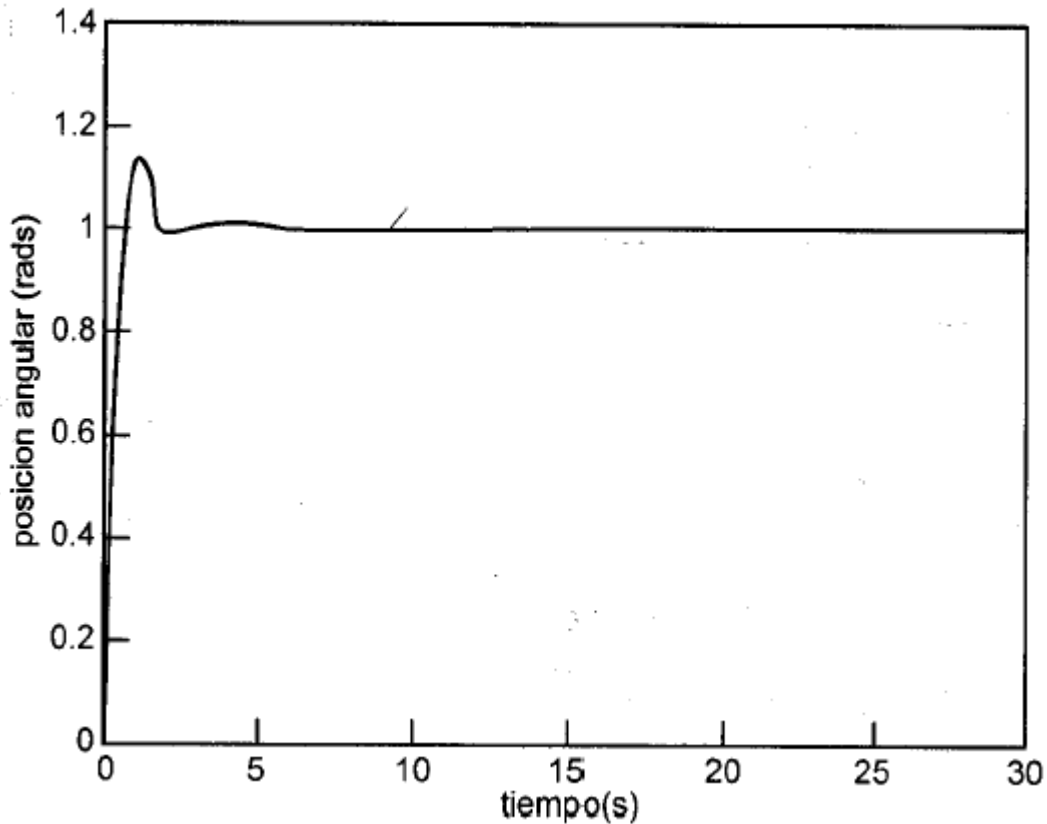


```

t=[0.0:0.01:30];
u=ones(1,length(t));
x0=[0;0;0;0];
y=lsim(AA,BB,CC,DD,u,t,x0);
plot(t,y)
end

```

Siguiendo con el ejemplo que se ha venido desarrollando y para un polo deseado adicional  $\mu_4 = -10$ , se obtiene la gráfica mostrada en la figura 11



**Figura 11: Respuesta temporal del servomotor realimentado controlado por armadura,.**

### **5.MODELACIÓN DE UN SERVOMECANISMO CONTROLADO POR CAMPO. [7],[8],[9]**

Cuando se trata de un motor dc controlado por campo, la corriente de armadura se mantiene constante. Recordando esto, la Ecuación 1 puede describirse de la siguiente manera:

$$T_m(t) = K_f i_f(t) \quad (19)$$

Ahora debe considerarse la malla del campo. Evaluando esta malla surge la expresión:

$$e_f(t) = R_f i_f + L_f \frac{di_f(t)}{dt} \quad (20)$$

El nuevo sistema de ecuaciones se completa con la Ecuación 8 (ecuación de torque en el eje), que sigue aplicando para el tipo de control que ahora se trata.

Las variables de estado  $x_1 = \theta$  y  $x_2 = \omega$  se eligen igual que en el caso anterior; pero en esta ocasión  $x_3$  será la corriente de campo  $i_f(t)$ ; y la entrada  $u$ , será la tensión de campo  $e_f(t)$ , aplicada al devanado de campo. Luego, manipulando las ecuaciones anteriores e introduciendo las definiciones nombradas:

$$\begin{array}{l} x_1 = \theta \Rightarrow \dot{x}_1 = \omega \\ x_2 = \omega \Rightarrow \dot{x}_2 = \alpha \\ x_3 = i_f \Rightarrow \dot{x}_3 = \frac{di_f}{dt} \\ u = e_f \\ y = \theta \Rightarrow y = x_1 \end{array} \quad \left| \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{K_r}{J}x_1 - \frac{B}{J}x_2 + \frac{K_f}{J}x_3 \\ \dot{x}_3 = -\frac{R_f}{L_f}x_3 + \frac{1}{L_f}u \\ y = x_1 \end{array} \right.$$

Obsérvese que la constante  $K_t$  ha sido sustituida por  $K_f$ ; nótese además la simplicidad con que se relaciona a la tercera variable de estado con la entrada.

El modelo en espacio de estados es el que se muestra a continuación:

$$\dot{\underline{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{K_r}{J} & -\frac{B}{J} & \frac{K_f}{J} \\ 0 & 0 & -\frac{R_f}{L_f} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L_f} \end{bmatrix} \underline{u}$$

$$\underline{y} = [1 \quad 0 \quad 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Como puede observarse, las matrices del sistema son prácticamente idénticas a las obtenidas para el Control de Armadura, a excepción de algunos detalles: la constante  $K_t$  es sustituida por  $K_f$ , la inductancia  $L_m$  se ve

reemplazada por  $L_f$ , y la resistencia  $R_m$  cambia por  $R_f$ . Adicionalmente, el elemento definido antes por  $K_e$  ahora es nulo.

Todos los comandos de Matlab descritos para el caso de Control de Armadura se aplican de la misma forma en este caso.

## 6. ADICIÓN DE UN OBSERVADOR DE ESTADO EN EL DISEÑO DE SISTEMAS SERVO TIPO 1.

El diseño de un sistema de control de seguimiento y el de un observador de orden completo, son independientes. El objetivo es calcular la matrices de realimentación  $K$  y  $K_e$  que se muestran en el diagrama de bloques de la figura 12.

Para el cálculo de la matriz  $K$  se procede del modo ya descrito en la sección correspondiente a servosistemas tipo 1 y RLE.

Para diseñar el observador de estado completo se requiere calcular previamente la matriz de observabilidad  $N$ . Es necesario aclarar que el comando `obsv` de Matlab calcula la traspuesta de la matriz  $N$  definida en la ecuación 14, por lo que habrá que tener esto en consideración al momento de utilizarla.

Para iniciar este cálculo se requiere la definición de los tres polos deseados para el observador ( $\mu_{o1}$ ,  $\mu_{o2}$  y  $\mu_{o3}$ ). Esta definición se hace en la diagonal de una matriz a partir de la cual se obtendrá el polinomio característico deseado, tal y como se hizo previamente para RLE. La secuencia de comandos de Matlab es:

$$J_o = [\mu_{o1} \ 0 \ 0; \ 0 \ \mu_{o2} \ 0; \ 0 \ 0 \ \mu_{o3}]; \text{ Poly}(J_o)$$

Para la aplicación de la fórmula de Ackermann, se necesita un polinomio matricial ( $\phi$ ).

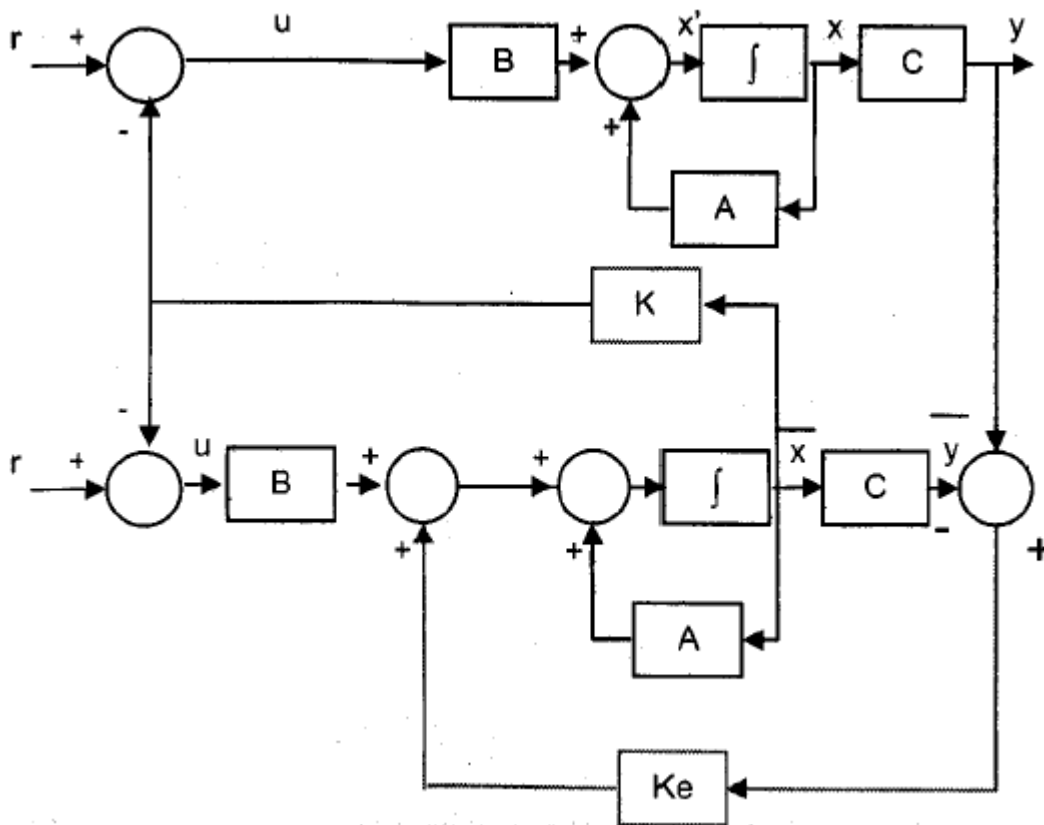


Figura 12: Sistema de seguimiento con realimentación de estado observado

```
Phio=polyvalm((Jo),A);
Ke=phio*(inv(N'))*[0;0;1]
```

Nótese que en la última línea, se usa la traspuesta la matriz N. Esto, conforme a la definición establecida e la ecuación 13; pero si se ha empleado el comando obsv para determinarla, no es necesaria esta trasposición.

```
At=[A-B*K B*K;zeros(size(A)) A-Ke*C];
Bt=[B;zeros(size(B))];
Ct=[C zeros(size(C))];
%gráfico
t=[0:0.01:30];
u=ones(1,length(t));
x0=[0;0;0;0;0;0];
y=lsim(At,Bt,Ct,d,u,t,x0);
```

Como paso previo a los comandos propios del gráfico se obtienen las matrices At, Bt y Ct. Ellas corresponden la simplificación del diagrama de bloques de la figura 12

Las líneas restantes ya son conocidas por su descripción previa. Únicamente resulta interesante comentar el tamaño del vector x0 de condiciones iniciales. Este tiene seis columnas, de acuerdo con el tamaño de la matriz At.

Puede verificarse como mejora la respuesta transitoria del sistema, con un observador realimentado y para uno polos deseados:  $\mu o1 = -2+3.4i$ ,  $\mu o2 = -2-3.4i$ ,  $\mu o = -10$ . Sin embargo, para obtener una correcta representación gráfica debe aplicarse un factor de escalamiento en el cálculo de la matriz Bt El lector interesado en aclarar este tópico particular puede referirse a la dirección siguiente: [www.engin.umich.edu/group/ctm/state/state.html](http://www.engin.umich.edu/group/ctm/state/state.html)

## 7. CONCLUSIONES.

El modelación de servomecanismos usando Matlab consiste en el uso de un grupo especializado de comandos que se aplican sobre las matrices de la ecuación de estado, definidas a partir de los parámetros eléctricos y mecánicos del servomotor.

Este procedimiento de cálculo, al ser repetitivo, permite aplicar una metodología de programación de tal forma que el usuario solo debe preocuparse por introducir correctamente los datos y de analizar los resultados obtenidos en las curvas que proporciona el software desarrollado.

Matlab ofrece posibilidades para diseñar una interfaz gráfica que facilita la interacción con el usuario, de modo: que este no requieran conocimientos detallados acerca de la teoría de espacio de estados para observar las curvas: que describen el comportamiento transitorio del sistema. Una mayor profundización sobre el desarrollo de esa interfaz' puede ser una continuación de este trabajo.

## 8. REFIIRENCIAS

1. MESA, W. "Introducción a los sistemas de Control Automático". Universidad de Carabobo. 1990.
2. CARNEGIE M. "Example: Root Locus Design for Digital DC Motor Position Control". [en línea]. Control Tutorials for Matlab. 24 08 1997, <http://www.engin.umich.edu/group/htm> [Consulta: 25 Octubre
3. CARNEGIE M. "DC Motor Position Control". [en línea]. Control Tutorials for Matlab. 12 08 1997, <http://www.engin.umich.edu/group/htm> [Consulta: 25 Octubre
4. CARNEGIE M. "Example: Modeling DC Motor Position". [en línea]. Control Tutorials for Matlab. 24 08 1997, <http://www.engin.umich.edu/group/ctm> [Consulta: 25 Octubre 2000].
5. CARNEGIE M. "Example: A State-Space Controller for DC Motor Position Control". [en línea]. Control Tutorials for Matlab. 22-08-1997 , <http://Www.engin.umich.edu/group/htm> [Consulta: 25 Octubre
6. OGATA, K. Ingeniería de Control Moderna. Editorial PRENTICE - HALL. Tercera Edición. 1998.
7. RORHS, C., MELSA, J. MELSA, J. y SCHULTZ, D., "Linear Control Systems". McGraw-Hill.
8. D'AZZO, J., "Linear Control system analysis and design"., New York: McGraw Hill, 1981
9. KAILATH, T., "Linear Systems", Englewood Cliffs, N.J.: Prentice Hall, 1985.