

ALGORITMO PARA EL CONTROL DE MOVIMIENTO BOBINO

Noé Eugenio Simons Langarica, José Rivera Mejía.
Instituto Tecnológico de Chihuahua.

Ave. Tecnológico # 2909 Colonia 10 de Mayo
C.P. 31310 Chihuahua, Chih.

nsimons@itchihuahua.edu.mx , jrivera@itchihuahua.edu.mx

RESUMEN

En el marco de inteligencia artificial y en particular el de sistemas difusos, se pretende realizar una aplicación en el contexto de la ganadería de un modelo de arreo como sistema experto.

En el presente artículo se exponen las ideas del desarrollo de un algoritmo para el control de movimiento de ganado desde una perspectiva práctica.

El algoritmo se programa en MatLab™. Con la información proporcionada por este, se plantean posibles escenarios para hacer una simulación de la trayectoria del bobino. Finalmente se presentan los resultados obtenidos y sus conclusiones.

1. INTRODUCCIÓN

Una tarea compleja, que además conlleva importantes costos, en una de las actividades más trascendentes de la historia reciente del estado de chihuahua es el arreo, ya que contempla la interacción y control de bovinos.

A pesar del avance de la ciencia y la tecnología, son recientes las investigaciones en las cuales se modela el movimiento o se implementen sistemas de control para el movimiento de animales. Entre los trabajos relacionados en la materia podemos mencionar el de [1], en el cual desarrolla un modelo de perro pastor para reunir patos. Lo realiza por medio de un control que procesa la información de una cámara de video la cual sigue y ubica a los patos.

Con otro enfoque, pero esencialmente para el mismo propósito tenemos a [2], el cual desarrolló un sistema para reunir ganado, instalando en los bovinos un dispositivo que estimula el sentido auditivo cuando el animal se está acercando a una zona prohibida. A este dispositivo se le llama cerca virtual, ya que realiza las funciones de una cerca convencional pero sin barreras físicas. La cerca virtual está basada en un receptor satelital,

el cual determina la ubicación permitida para el bobino.

Un análisis minucioso de las cercas virtuales, donde además se exponen las ventajas, oportunidades, problemas principales y además las características de algunas patentes se desarrolla en [3]. Empero [3], enfatiza en la necesidad de investigar, mejores esquemas de adaptación del animal al estímulo aversivo.

2. DESARROLLO

En esta sección se presenta la metodología aplicada en la investigación, la cual tiene como objetivo proponer un algoritmo práctico que permita ayudar en la tarea de arreo, hacer más eficiente la ganadería extensiva y reducir el daño ecológico que genera esta actividad.

Como primer paso se realizaron 18 encuestas a ganaderos tomados al azar, que acudieron a realizar trámites en la unión ganadera de Chihuahua, con el fin de entender como la actividad milenaria, el arreo, se efectúa actualmente y así determinar las características para un sistema experto.

Todos los sujetos entrevistados contestaron que si habían arreado ganado y se valían de caballos para hacerlo, solo algunos utilizaban perros y otras herramientas. Solo 2 encuestados respondieron que el arreo se debería realizar en silencio, el resto que utilizaba alguna clase de sonido. Así mismo el 85% dijo que el arreo debería ser lento. También se encontró que dedicaban en un promedio de 3 horas a la semana para esta actividad y que entre 3 vaqueros arrean un promedio de 200 vacas. 4 contestaron que conocían el término de cerca virtual y 5 que no invertirían en algún equipo que les ayudara a realizar el arreo más fácilmente.

En el contexto de investigación científica recabada sobre el comportamiento bobino y con la valoración de la información en las encuestas se procedió a realizar el siguiente paso.

2.1. Planteamiento del problema.

Para fundamentar el problema se realiza el siguiente planteamiento. Sean $U_f, U_i \in R^2$, donde U_i es un punto en el plano (x, y) que representa la ubicación del bobino en un determinado tiempo y U_f la ubicación fija deseada, entonces necesitamos encontrar un valor $\varepsilon \in R$ tal que $\|U_f - U_i\| \leq \varepsilon$.

Para tal efecto se considero la hipótesis de que es posible incidir en el movimiento de un bobino por medio de señales auditivas y visuales, ya que dichos angulados tienen ampliamente desarrollados sus sentidos (vista y oído) [4].

2.2. Modelo para el arreo de bovinos.

Para el modelado se consideraron algunas alternativas, tales como: modelos ocultos de Markov, variables de estado o las ventajas de un sistema neuro-difuso, [5]. Por las circunstancias que imperan para la implementación, se opto por las facilidades de un sistema difuso, que permitiera manejar la comodidad de un modelo impreciso pero eficaz.

En la Fig. 1, podemos apreciar como a través de una serie de vectores de posición se modela los movimientos del bobino. Estos vectores son muestreados por puntos que representan la ubicación del bobino en un instante determinado. Se observa un vector de trayectoria el cual representa el rumbo deseado.

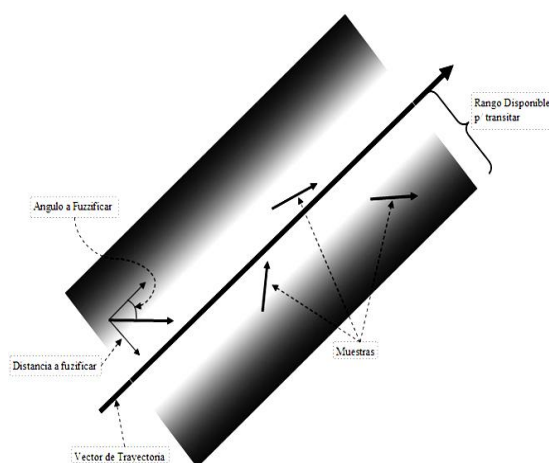


Figura 1. Interacción de vectores en la zona de tolerancia del bobino.

Los vectores muestra, son los que serán analizados por el sistema de control difuso y así tomar la decisión sobre el tipo de señal que se enviara al bobino de acuerdo a la zona de tolerancia o disponible para navegar. En la Fig. 1, se muestra también que la zona cercana al vector de trayectoria deseado seria estimulado con sonidos no estresantes. Por otra parte la zona oscura seria estimulado con sonidos estresantes y alarmantes. El ángulo y el modulo de una componente de los vectores muestras, serán la información que se procesara en el control difuso.

Como es típico en la práctica, existe una zona de protección en la cual el bobino buscara mantenerse. En la Fig. 2 se aprecia cómo funciona lo que se denomina zona de fuga investigada por [4]. Esta nos ayuda a entender como, el animal es motivado a moverse por la presencia cercana de alguna amenaza.

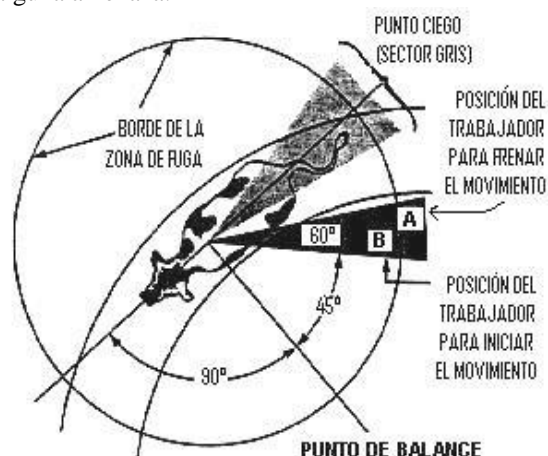


Figura 2. Partes del diagrama de zona de fuga.

Finalmente en la Fig. 3. se ilustran los componentes que conforman las etapas del proceso para control de movimiento del bobino. Y se observa un esquema típico de lazo cerrado con sensores que monitorean la ubicación del animal, la comparan con la deseada, se genera un estímulo y el proceso continua hasta lograr un error del 5 %.

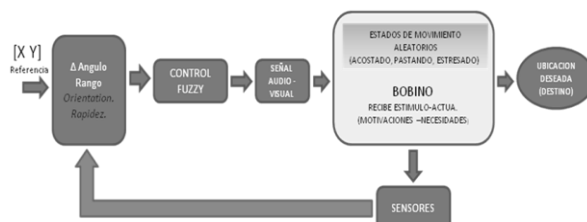


Figura 3. Diagrama del proceso de control.

2.3. Simulación.

Como punto de inflexión entre la etapa de diseño y la implementación se hace una simulación en este caso mediante MatLab [6].

En la Fig. 4. se aprecian las modificaciones de la Fig. 3, para la simulación en MatLab.

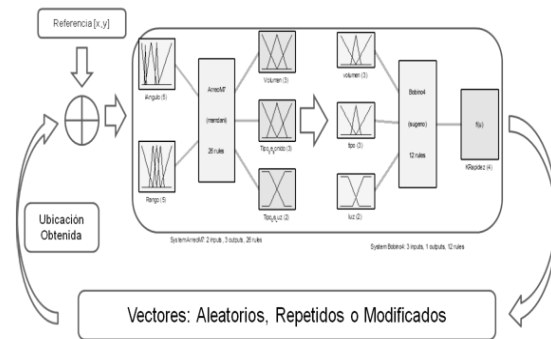


Figura 4. Arquitectura del simulador.

La parte del control difuso consta de dos entradas (iAngulo y Rango) y tres salidas que serán evaluadas en el sistema bobino, Fig. 5.

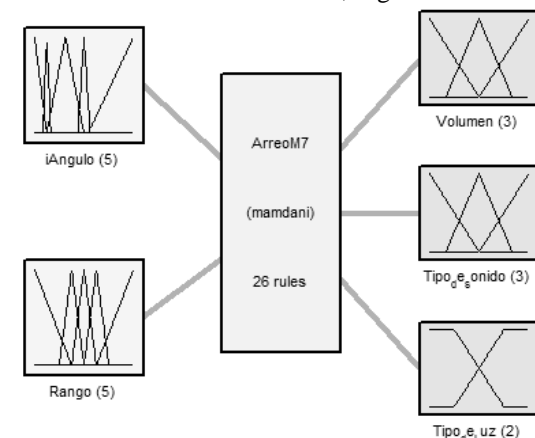


Figura 5. Sistema difuso para el control.

En el modelo del bobino, de la Fig. 6, se aprecia un sistema difuso de tres entradas y una salida, que representa la rapidez con la que actúa el bobino a los distintos estímulos generados en el control difuso.

Las 3 entradas del modelo son: Volumen y tipo de sonido y tipo de luz, que son las señales que percibirán sus sentidos (oído y vista). De acuerdo a estos estímulos, se genera el parámetro KRapidez para determinar cómo será el movimiento que realizará el bobino. Una vez que se conoce KRapidez, el simulador genera el vector

de movimiento del bobino, que se realimenta y se compara con el vector deseado.

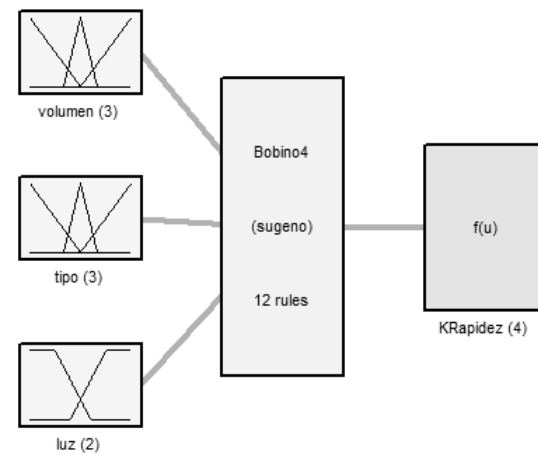


Figura 6. Sistema difuso para modelo de bobino.

Los vectores que complementan el modelo difuso del bobino son generados considerando las características de un animal manso tales como: 1) Rapidez de referencia 2 m/s, [7], 2) El movimiento de los animales es directamente proporcional a los estímulos auditivos y visuales, 3) El animal no camina en línea recta.

En la fig. 7, observamos el efecto que tiene el modelo del bobino a diferentes niveles de volumen y tipo de sonido. Algunos ejemplos de sonidos tipo 2 (alarmante) y 3 (estresante) son, ladridos de perro y las altas frecuencias respectivamente.

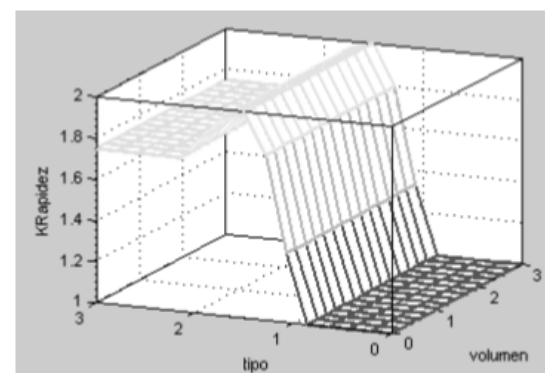


Figura 7. Modelo difuso para el bobino.

El simulador genera los escenarios de la siguiente manera: Si el Rango es Malo, Fig. 8, el control envía estímulos estresantes, los estímulos recibidos del control generan una salida $KR_{rapidez} \approx 3$ (alta) y se produce un vector aleatorio, que

significa que el animal llega al límite de su frontera y se le está indicando que modifique su dirección, pero no sabemos con certeza hacia donde lo hará. Esta acción se repite, como se muestra en la Fig. 5, hasta que se detecte un cambio en la dirección correcta por $i\text{Angulo}$ y una mejoría en el Rango permitido. Otro escenario hipotético sería, $K\text{Rapidez} \approx 1$, entonces el sistema se queda sin cambios y genera un vector similar al anterior, lo que significa que el bobino no necesita modificar su dirección.

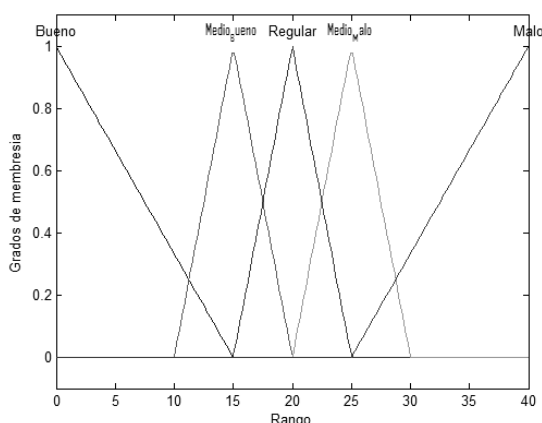


Figura 8. Variable Rango y sus funciones de pertenencia en el simulador.

De esta manera es posible modelar el comportamiento del movimiento de un bobino, de acuerdo a los estímulos recibidos, simularlo y estar en condiciones de implementarlo en campo.

2.4. Seudocódigo

El pseudocódigo de la Fig. 9, está planteado para una trayectoria simple, ya que este mismo proceso podrá ser usado cuando el trayecto se divida en dos o más tramos.

Como se aprecia en la Fig. 9, se calcula el vector base (rumbo deseado), posteriormente se muestrea la ubicación actual del bobino. Por medio de 2 muestras se calcula su dirección actual, la cual también predice la probabilidad de su rumbo futuro. Enseguida se determina el universo de discurso para el control difuso, tomado de comparar los vectores base y de dirección actual. Finalmente se procesa el sistema difuso y se obtiene los parámetros que serán utilizados para estimular al animal.

Algoritmo:

In: U_i, U_f, U_i ; // U_i, U_f , ubicación inicial y final.

Out: U_o ; // U_i, U_o , ubicación instantánea y de salida.

1. $V_b = U_f - U_i$ // Rumbo dado por el usuario.
2. Punto aleatorio inicial U_1 ; // Del GPS
3. Pausa;
4. Punto aleatorio U_2 ;
5. **While** $|U_f - U_o| < 5\%$;
6. Vector $V_1 = U_i - U_1$;
7. $V_2 = U_i - U_2$;
8. $V_{1_2} = V_2 - V_1$;
9. $i\text{Angulo} \leftarrow \text{Angulo } V_b, V_{1_2}$;
10. $\text{Rango} \leftarrow V_2 - \text{Proy } V_2 / V_b$;
11. Fuzzificar $i\text{Angulo}$ y Rango ;
12. R1: **If** $i\text{Angulo}$ is -120 and Rango is bueno **then** Sonido is estresante;
13. R2: **if** $i\text{Angulo}$ is 0 and Rango is medio **then** Sonido is alarmante;
14. R3:...
15. Defuzzificar para obtener Tipo de sonido, Nivel, Tipo de Luz;
16. Estimular al animal;
17. **End**;

Figura 9. Algoritmo.

3. PRUEBAS Y RESULTADOS.

En la Fig. 10 se muestran los resultados para una trayectoria simple. En ella se observa que existe convergencia hacia el objetivo, en este caso el rectángulo.

También podemos ver que el algoritmo no termina exactamente su último punto sobre el rectángulo y es porque el modelo maneja una tolerancia de ± 20 mts. Esta tolerancia es permitida ya que, un ganadero podría ver si un bobino llega a una distancia de al menos 20 mts.

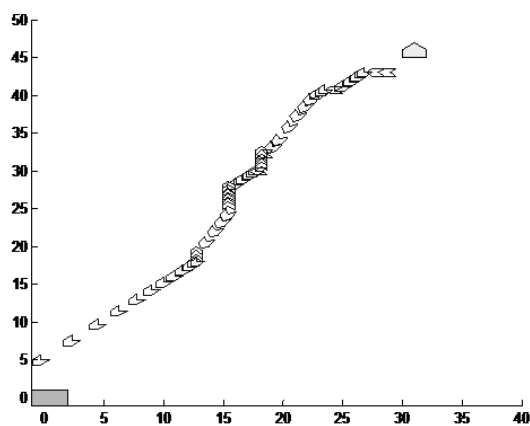


Figura 10. Resultado de una trayectoria simple.

En la Fig. 11 se muestran los resultados en un área más amplia. Aquí se programaron tres puntos los cuales el algoritmo tenía que recorrer. Este debía partir de (50,300), luego llegar al punto (150,100), posteriormente ir a (0,0) y finalmente terminar en donde empezó (50, 30). Esta podría ser una trayectoria práctica cuando se requiere evadir algún obstáculo, como una siembra de maíz por ejemplo.

Es importante apreciar que las trayectorias son muy similares en su forma general, sin embargo tienen algunas diferencias debido a que cada trayectoria es generada por vectores aleatorios, que son procesados por el control difuso para cumplir con la convergencia hacia el objetivo planteado.

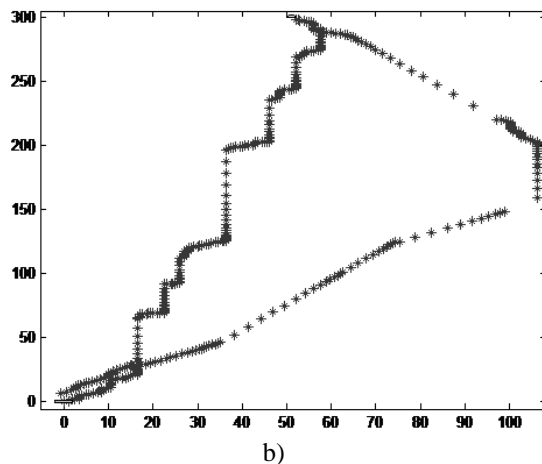
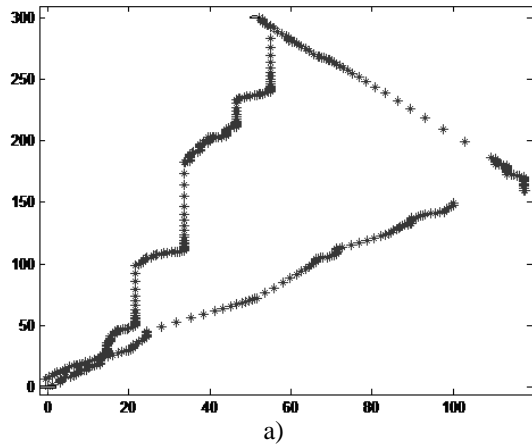


Figura 11. Patrón de trayectoria con múltiples tramos.

a) Prueba 1: el simulador cumple con la trayectoria planeada.

b) Prueba 2: cumple con la trayectoria y se aproxima a la prueba 1, sin embargo, por los vectores aleatorios las trayectorias son diferentes.

4. CONCLUSIONES.

Este trabajo se desarrolló con el propósito de contar con un algoritmo práctico, que pueda ser implementado en un sensor inteligente, [8], montado sobre el animal.

Las pruebas realizadas, validan su eventual implementación en hardware. Es importante mencionar que es necesario también, un análisis de confiabilidad del sistema, antes de instalarse en el bobino por si algún mecanismo falla.

Además podemos considerar que, este algoritmo una vez implementado en un sensor inteligente, sería factible, experimentar con distintos tipos de actuadores, combinarlos con algunas clases de angulados y ver que actuadores son más factibles para cada caso.

5. BIBLIOGRAFÍA.

- [1] R. Vaughan, N. Sumpter, J. Henderson, A. Frost, S. Cameron, «Experiments in automatic flock control,» *Robot Auton Syst*, vol. 31, n° 1-2, pp. 109-117, 2000.
- [2] Z. Butler, P Corke, R. Peterson, D. Russ, «From robots to animals: Virtual Fences for Controlling Cows,» *Int. J. Robot Res*, vol. 25, n° 5-6, pp. 485-508, 2006.
- [3] C. Umstatter, «The evolution of virtual fences: A review,» *Comput Electron Agr*, vol. 75, n° 1, pp. 10-22, 2011..
- [4] Grandin, T, «Behavioral Principles of Livestock Handling,» [En línea]. Available: <http://www.grandin.com/references/new.corral.html>. [Último acceso: Jul 2011].
- [5] Lefteri H. Tsoukalas, Robert E. Uhrig, «Neural Networks and Fuzzy Logic Systems,» de *Fuzzy and Neural Approaches in Engineering*, New York, John Wiley & Sons, 1997, pp. 1-3.
- [6] MatLab, [En línea]. Available: [www.mathworks.com/products/matlab/..](http://www.mathworks.com/products/matlab/) [Último acceso: Enero 2012].
- [7] Von Pless, G. Butler, Z., «Adaptive expert systems for indirect coverage control,» de *Robotics and Automation, 2009. ICRA '09. IEEE International Conference*, pp. 1724-1730, Kobe, 2009.
- [8] Randy Frank, «Smart sensors basics,» de *Understanding Smart Sensors*, Norwood, Artech House, 2000, pp. 3-3.