

Enseñanza-aprendizaje constructivista a través de la Robótica Educativa

Javier Arlegui de Pablos

Departamento de Psicología y Pedagogía

Alfredo Pina Calafi

Departamento de Ingeniería Matemática e Informática

Universidad Pública de Navarra

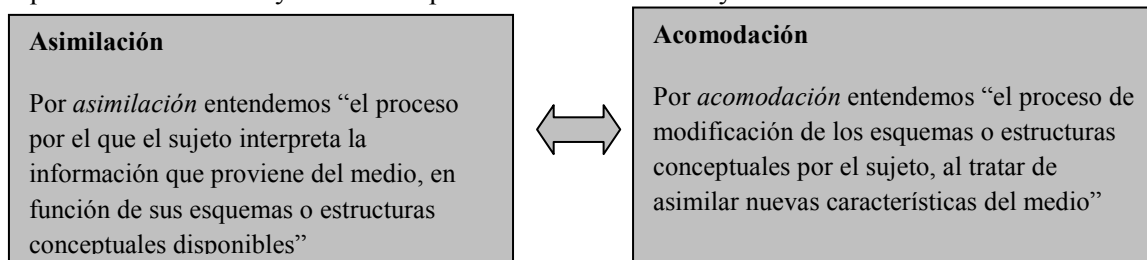
1.- Una propuesta para la enseñanza – aprendizaje constructivista en robótica

La competencia de “ordenar tareas a robots para que realicen determinados comportamientos (con un objetivo en mente)” puede ser el objeto de una enseñanza (por parte del profesor) y un aprendizaje (por parte del alumno) constructivista.

Debemos, para ello, seleccionar y adaptar a nuestro objetivo los rasgos más pertinentes de las teorías de Piaget [1][2] y Vygotskii [3][4], conocidas como teorías de la reestructuración cognitiva y que asumen una enseñanza-aprendizaje constructivista.

1.1.- El proceso de adaptación, según Piaget

La teoría de Piaget es una teoría de la *construcción dinámica* del conocimiento. Piaget basa dicha construcción en el proceso de *adaptación “mayorante”*, que formula como la tendencia a un equilibrio cada vez mayor entre los procesos de *asimilación* y *acomodación*.



La asimilación sugiere que “vemos” las cosas no como son, sino como somos nosotros. Las vemos según nuestros esquemas de comprensión disponibles. Incorporamos de la realidad sólo aquellos elementos “inclusores” que reconocen nuestros esquemas previos (véase al respecto la teoría (constructivista) de AUSUBEL [5] sobre el aprendizaje verbal significativo). Si únicamente existiera la asimilación, gran parte de nuestros conocimientos serían fantásticos y conducirían a continuas equivocaciones.

La acomodación explica la tendencia de nuestros esquemas de asimilación a adecuarse a la realidad, y a irse transformando en esquemas más “acordes” (o más *equilibrados*) con ella. Si mis esquemas son insuficientes para asimilar una situación determinada, probablemente modificaré alguno de mis esquemas adaptándolo para interpretar características adicionales de la situación.

Pero la acomodación supone no sólo una modificación de los esquemas previos en función de la información asimilada, sino también una nueva asimilación o reinterpretación de los datos o conocimientos anteriores en función de los nuevos esquemas construidos. Es lo que denominamos “reestructuración” y es el efecto más importante de la adaptación y del propio proceso constructivista..

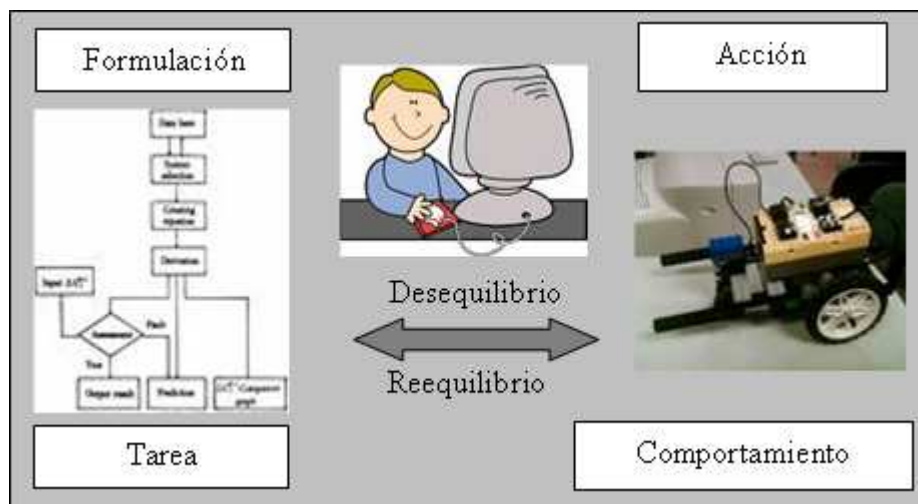


Figura 1: Proceso de desequilibrio/reequilibrio en el ámbito de la robótica educativa

1.2.- La enseñanza – aprendizaje como un proceso de reequilibración sucesiva

Cuando un alumno inicia un primer contacto con una nueva “obra de saber”¹ está generalmente desequilibrado frente a ella. Le aplica sus esquemas previos (su contexto) cognitivo y generalmente asimila sólo en parte aspectos del objeto. Un doble trabajo: de *interacción empírica* directa con el objeto y de *interacción lingüística* con un profesor (con referencia al objeto), le facilitará al alumno una progresiva adaptación tanto a la comprensión de las acciones de dicho objeto cuanto a la comprensión de los términos del lenguaje con el que describimos dichas acciones.

Ello le llevará a una primera reequilibración, pero una nueva interacción con el objeto Y/o una nueva pregunta problemática del profesor respecto al objeto, llevará al alumno a una nueva situación de desequilibrio que deberá superar por el mismo procedimiento (de hacer-con y de hablar-sobre el objeto) para alcanzar un nuevo estado de equilibrio “mayor” que el anterior, en el sentido que permite una mayor comprensión de las características socio-naturales del objeto. Por ello se habla Piaget de una adaptación “mayorante”. Y así sucesivamente...

El papel del profesor es provocar sucesivos desequilibrios “controlados” (no introducir demasiados aspectos nuevos en cada pregunta) y, luego ayudar, guiando con “mostraciones” (al estilo de lo que plantea Vygotskii [3][4]) al reequilibrio del alumno (mostrar acciones contingentes con el objeto “bien realizadas”, y expresiones lingüísticas con referencia al objeto contingentes y “bien formuladas”).

El papel del alumno es esencialmente seguir intelectualmente “activo” en el proceso, esforzándose en identificar e incorporar los nuevos inclusores en los esquemas previos y tratando de dotar de significación las mostraciones del profesor y construir con él la elaboración de significados compartidos.

¹ sirva este término para designar cualquier elemento, más o menos complejo, de nuestro entorno cuya estructura y/o función debe ser *explicada* posteriormente por el alumno

Digamos, finalmente, que frente a las “obras de saber”, donde el alumno debe poder referirse verbalmente a ellas o sobre ellas, no cabe el constructivismo individual del alumno; se trata necesariamente de un constructivismo “guiado” del alumno: interactuando con el objeto y, simultáneamente, dialogando con el profesor.

1.3.- Niveles de complejidad de la reequilibración

Siguiendo a J. I. Pozo [6], Piaget elaboró, a lo largo de su obra, varios modelos del proceso de equilibración. En el último de ellos sostiene que el equilibrio entre asimilación y acomodación se produce (y se rompe) en tres niveles de complejidad creciente:

- En un primer nivel: *equilibración con los hechos*:
Los esquemas del individuo deben alcanzar el equilibrio con los nuevos objetos que asimilan.
- En un segundo nivel: *equilibración con los esquemas*:
Tiene que alcanzarse un equilibrio entre los viejos y nuevos esquemas del individuo, que deben asimilarse y acomodarse recíprocamente.
- En un tercer nivel: *equilibración con la estructura jerárquica de esquemas*:
Tiene que alcanzarse un equilibrio entre las viejas y nuevas jerarquías de esquemas.

2.- Una propuesta para la enseñanza – aprendizaje constructivista en robótica

2.1.- Los problemas

La enseñanza de la robótica debería proceder por la *formulación sucesiva de problemas, por el profesor*, relativos a un mismo Tema.

Se trata de:

- *Problemas sobre una misma clase de comportamientos* (acciones del robot), lo que debe dar origen a una *misma clase de tareas* (de programación), de las cuales debe surgir el “buen” *procedimiento técnico* (una “técnica”) de programación. Este procedimiento es común para toda la clase de tareas y ejecuta cada una de ellas por asignación de sus parámetros específicos.
- *Problemas sobre una nueva clase de comportamientos*, próxima a la anterior y que se solapa en parte con la secuencia anterior de problemas

2.2.- Unidades didácticas constructivistas

De todo lo anterior podemos permitirnos separar el proceso de la construcción de *un curso constructivista* en dos componentes:

- *La tarea del profesor*, que se concreta en el establecimiento de los problemas-tipo y la secuenciación de los mismos en los “recorridos de estudio e investigación” que propondrá a los alumnos.
- El profesor se ocupará, así, de:
 - la definición/transposición y proposición de los problemas,
 - las *regulaciones y mostraciones* en el acompañamiento “vygotskyano” del aprendizaje de sus alumnos
 - la *institucionalización* de los saberes emergentes de los alumnos
- La tarea del(los) alumno(s), que se concreta en el esfuerzo de construcción indagatoria de conocimiento relativo a los saberes en juego. Un conocimiento significativo y funcional, que requiere la construcción de unidades praxeológicas de “saber-hacer”.

2.3.- Etapas de una unidad didáctica constructivista

a.- *Formulación, por el profesor, de un problema-tipo*

Es un problema referido no a un comportamiento determinado, sino a un tipo (una clase) de comportamientos deseables del robot.

La soluci n consistir  en buscar (*construir*) un *procedimiento* que proporcione la clase de tareas adecuada para la clase de comportamientos deseada

Los modernos trenes autom ticos (sin conductor) se detienen tras recorrer distancias precisas entre estaciones.... Podr amos construir un robot que act e como un tren "inteligente"... Para ello:
 C mo hacer que un robot m vil recorra una distancia D precisa, previamente establecida...?

b.- Ejecuci n exploratoria, si es necesario, por los alumnos de las acciones pertinentes que el robot debe llevar a cabo para ejecutar soluciones particulares del problema

Con el fin de comprender mejor el propio problema, los alumnos (si lo necesitan) ejecutan manualmente las acciones correspondientes a soluciones particulares del problema. En un modo de pilotaje (o simplemente, mediante una simulaci n humana del robot) los alumnos remedan las acciones deseables para que el robot d  respuestas particulares al problema formulado (esta etapa es tanto m s necesaria cuanto m s j venes son los alumnos...)

Suponiendo en un principio la ejecuci n de recorridos lineales, s lo para ni os muy peque os ser  necesario una ejecuci n manual del recorrido de un m vil entre dos puntos A y B previamente establecidos (cuya distancia se conoce)...

c.- Formulaciones, por los alumnos, de soluciones particulares

Escribir exploratoriamente tareas (mediante instrucciones o secuencias de instrucciones) que hagan que el robot ejecute comportamientos variados de la clase que se desea

Un grupo de alumnos proponen utilizar tentativamente la instrucci n primitiva "motor", ya implementada en NXT-G, y deciden variar inicialmente el par metro "degrees" (dejando el par metro "power" fijo con un valor constante).

Preparan una serie de cuatro tareas (de un mismo tipo), se alando en el suelo cuatro recorridos lineales A-B con diferentes distancias cada uno: distancia_{1AB}= 50 cm; distancia_{2AB}= 100 cm; distancia_{3AB}= 150 cm; distancia_{4AB}= 200 cm

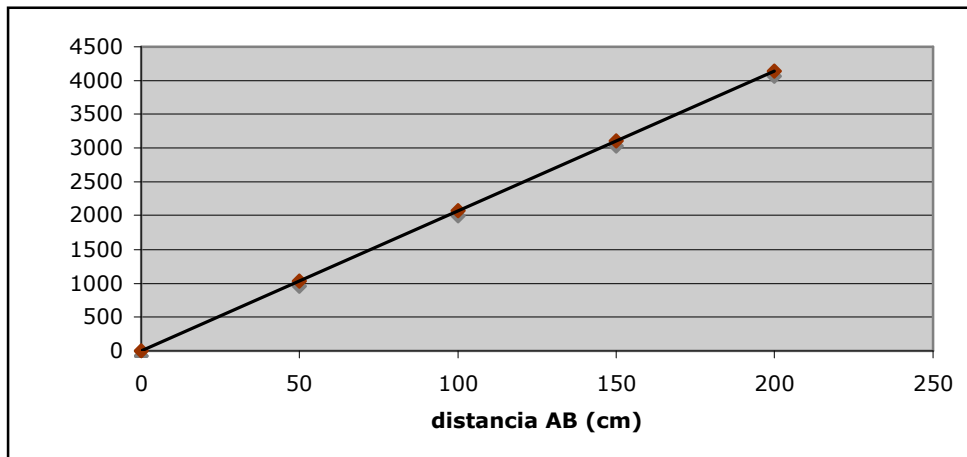
Y se plantean programar al robot para que realice cuatro tareas: recorrer exactamente esas cuatro distancias.

Comienzan programando, por ensayo y error, diversos recorridos del robot con diversos valores del par metro "degrees", hasta que, finalmente, consiguen que recorra exactamente (con un m nimo error aceptable) y de modo sistem tico la distancia_{1AB} de 50 cm (lo repiten varias veces). Eso ocurre siempre para un valor del par metro "degrees" de 1033°...

Siguen programando por ensayo y error hasta conseguir que el robot recorra exactamente las restantes distancias, obteniendo, finalmente los datos de la siguiente tabla para :

| tarea | distancia AB (cm) | degrees |
|-------|-------------------|---------|
| | 0 | 0 |
| 1 | 50 | 1033 |
| 2 | 100 | 2070 |
| 3 | 150 | 3106 |
| 4 | 200 | 4138 |

Realizan una descripci n gr fica de los datos, obteniendo:



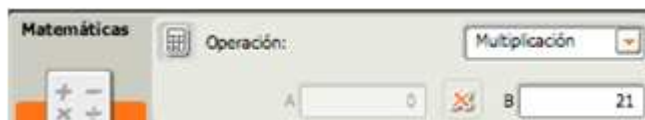
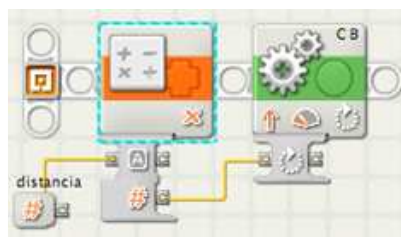
d.- Generalización de las soluciones: formulación de un procedimiento

Integración de las tareas en un saber técnico (una técnica) que se describa mediante un procedimiento parametrizado de programación que resuelva la clase de comportamientos

Se observa que la relación entre ambas variables es proporcional y establecen finalmente, con su propia expresión, que “el robot requiere un valor de 2070 degrees por cada cien centímetros que avanza (o un valor de 20,7 degrees por cada cm que avanza), y los “degrees” son proporcionales, con ese factor, a los centímetros que avanza”.

Con este “saber” que han construido, pueden ahora programar el robot de un modo general para que resuelva en todos los caso el problema planteado, y camine un determinado número de centímetros, que se introducen como dato directamente en el programa (ver diagrama adjunto con el procedimiento empleado)

Escritura “canónica” del procedimiento: con las dos instrucciones de la figura construyo un procedimiento denominado “*Avanza_distancia (distancia)*” que toma como parámetro el valor de la variable “*distancia*”



e.- Validación del mismo

Mediante la aplicación exitosa del procedimiento a nuevos casos (de la misma clase que los precedentes)

Se escribe la tarea *avanza una distancia* con el valor 40 para el parámetro distancia. . Se comprueba el comportamiento del robot: se mide la distancia que recorre y se confirma que se desplaza 40 cm con gran precisión. Es inmediata la comprobación del movimiento exacto del robot introduciendo nuevas distancias como datos.



f.- Generalización y extensión del procedimiento a nuevos problemas-tipo

Con el estudio de las limitaciones y el alcance del procedimiento obtenido se inicia su adaptación para la resolución de nuevos problemas-tipo...

¿Qué ocurre si cambiamos las ruedas del robot?... ¿Qué ocurre si hacemos variar simultáneamente el valor de Power en el motor?... estos son ejemplos de posibles “recorridos” de investigación constructiva que llevarían, por una secuencia semejante de pasos a plantearnos problemas de un “segundo nivel”, con procedimientos en función de dos variables y la construcción de saberes que constituyen leyes de carácter tecnológico.

3.- Un ejemplo: movimiento linear uniforme

En este ejemplo mostramos una secuencia “constructiva” (de la parte del profesor) para estudiar el comportamiento básico de un robot, partiendo de los ensamblajes más simples de robots y de las técnicas más elementales que pueden ejecutar.











Aunque ordenaremos realizar al robot acciones físicas (desplazamientos, giros, ...) nuestro interés primario no será “usar” al robot para conocer los aspectos físicos del entorno (dado que todavía no sabemos siquiera cómo “hablarle”, por el contrario, en estas etapas iniciales estamos más interesados en explorar los términos básicos de su lenguaje, eso sí, de un modo funcional, haciendo que haga con ellos acciones simples pero con significado en el entorno (desplazamientos y giros simples, etc).

Esto lo hacemos con lo que denominaremos una “secuencia de estudio e indagación” (por temas y secuencias de problemas) del robot LEGO MINDSTORMS “*como objeto de conocimiento*”. Concretamente en este ejemplo trataremos:

- el movimiento linear uniforme (*velocidad = espacio/tiempo*)
- aspectos básicos de construcción de robots (estabilidad, rigidez, etc....)
- aspectos básicos de programación del robot (variables, procedimientos, bucles, etc....)

3.1.- La secuencia de problemas

En la siguiente tabla podemos visualizar 6 diferentes problemas a resolver, donde se indica para cada uno de ellos los procedimientos formales que trabajaremos, tanto en pseudo-código como en el lenguaje gráfico NXT-G.

| | problema-tipo | procedimiento en pseudo-código | procedimiento en LEGO NXT-G |
|----------|---|---|---|
| 0 | ¿cómo ensamblar “buenos” robots lineales...? | | |
| 1 | ¿cómo ordenar a un robot simple que se desplace hacia delante con una cierta potencia durante un cierto intervalo t de tiempo ...? | Move_PT (power, time) |  Re-escribiendo obtenemos: MOTOR (power, time) |
| 2 | ¿cómo ordenar a un robot simple que se desplace hacia adelante con una cierta potencia a lo largo de una cierta distancia angular ϕ (de sus ruedas) ...? | Move_PD (power, degrees) |  Re-escribiendo obtenemos: MOTOR (power, degrees) |
| 3 | ¿cómo ordenar a un robot simple que se desplace hacia adelante con una cierta potencia a lo largo de una cierta distancia linear x (en cm) ...? | Move_PX (power, x) (transformamos distancias angulares (degrees) en distancia lineares (cm)) |   Operacion (x) -> degrees Y obtenemos: MOTOR (power, x) Re-utilizamos: Move_PD (power, degrees) |
| 4 | ¿cómo ordenar a un robot simple que se desplace hacia adelante con una cierta velocidad durante un cierto tiempo...? | Move_VT (v, time) |   Operacion (v) -> power Y obtenemos: MOTOR (v, time) Re-utilizamos: Move_PT (power, time) |
| 5 | ¿cómo ordenar a un robot simple que se desplace hacia adelante con una cierta velocidad y recorrer una cierta distancia linear x ...? | Move_VX(v,x) |   Operacion (x) -> time Y obtenemos: MOTOR (v, x) Re-utilizamos: Move_VT (v, time) |
| 6 | ¿cómo ordenar a un robot simple que se desplace hacia adelante a lo largo de una cierta distancia x durante un tiempo t...? | Move_XT(x,time) |   Operacion (x) -> v Obtenemos: MOTOR (x, time) Re-utilizamos: Move_VT (v, time) |

A partir de aquí otros problemas que se pueden plantear son:

- ¿cómo podemos ordenar un desplazamiento *grande* como una secuencia de desplazamientos *pequeños*...?
- ¿cómo realizar sucesivos pequeños desplazamientos en el tiempo pero que sean cada vez más grandes en longitud?
- ¿cómo combinar el desplazamiento con otra actividad ¿

Al hacerlo preparamos la introducción de bucles y otras estructuras de control así como otros aspectos relacionados con los contenidos como la aceleración.

3.2.- El desarrollo de los problemas

Tal y como hemos comentado anteriormente para cada problema planteado hay diferentes apartados a resolver como la formulación del problema, sus objetivos, saberes a adquirir y saberes previos requeridos, o sugerencias. Vamos a dar algunos detalles de dos de los problemas (P1 y P4, que utiliza en parte lo conseguido en P1). Consideramos que ya hemos hecho el problema 0, la construcción del robot.

3.2.1.- Problema 1: Move_PT (power, time).

Formulación del problema: ¿cómo ordenar a un robot simple que se desplace hacia delante con una cierta potencia (power) durante un cierto intervalo t de tiempo ...?

Objetivos: Que los alumnos sean capaces de escribir y de usar un procedimiento tipo comando que desplace linealmente un robot durante un intervalo de tiempo t y con una potencia dada

Saberes a adquirir y saberes previos requeridos:

Saberes a adquirir:

- Lenguaje NXT: bloque Move, uso de variables y creación de procedimientos
- Iniciar el “recorrido constructivista” creando bloques básicos que se reutilizaran mas adelante

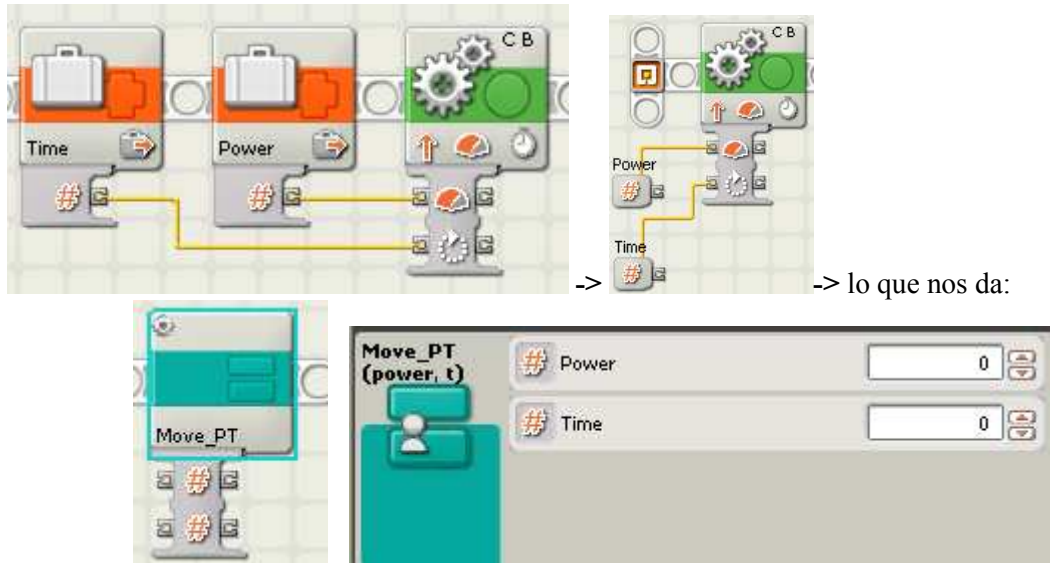
Saberes previos:

- ninguno sobre procedimientos NXT-G (en esta unidad se introducen los primeros, bloques, variables y procedimientos del lenguaje de programación),
- los de la unidad anterior: la construcción de un robot lineal,
- conocimiento básico del entorno de programación NXT-G: edición, comunicación con el robot, cómo bajar programas, etc

Originalmente, el bloque de control de motores, Move tiene los siguientes parámetros:



Este primer problema tiene como objetivo básico “simplificar” este bloque, reescribiéndolo como una procedimiento en el cual controlamos como máximo 2 parámetros. El resultado es:el nuevo loque Move_PT con 2 parámetros unicamente:



3.2.2.- Problema 4: Move_VT (v, time).

Formulaci3n del problema:  C3mo podemos convertir los valores de velocidad “v” [cm/s] en valores de “power”?

Objetivos: Que los alumnos sean capaces de escribir y de usar un procedimiento tipo comando que desplace linealmente un robot durante un intervalo de tiempo t y con una velocidad dada

Saberes a adquirir y saberes previos requeridos:

Saberes a adquirir:

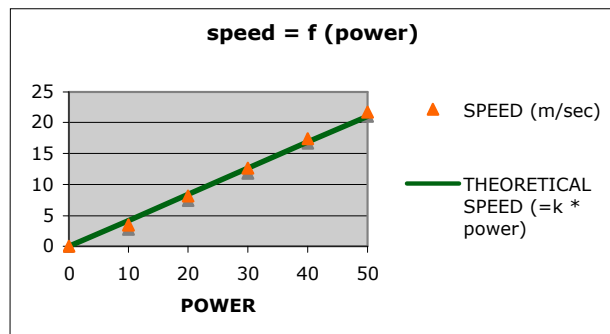
- Crear un bloque Move_VT con 2 parametros, velocidad (cm/seg) y tiempo (seg)

Saberes previos:

- Queremos que se utilice el bloque anterior, Move_PT (power, time)

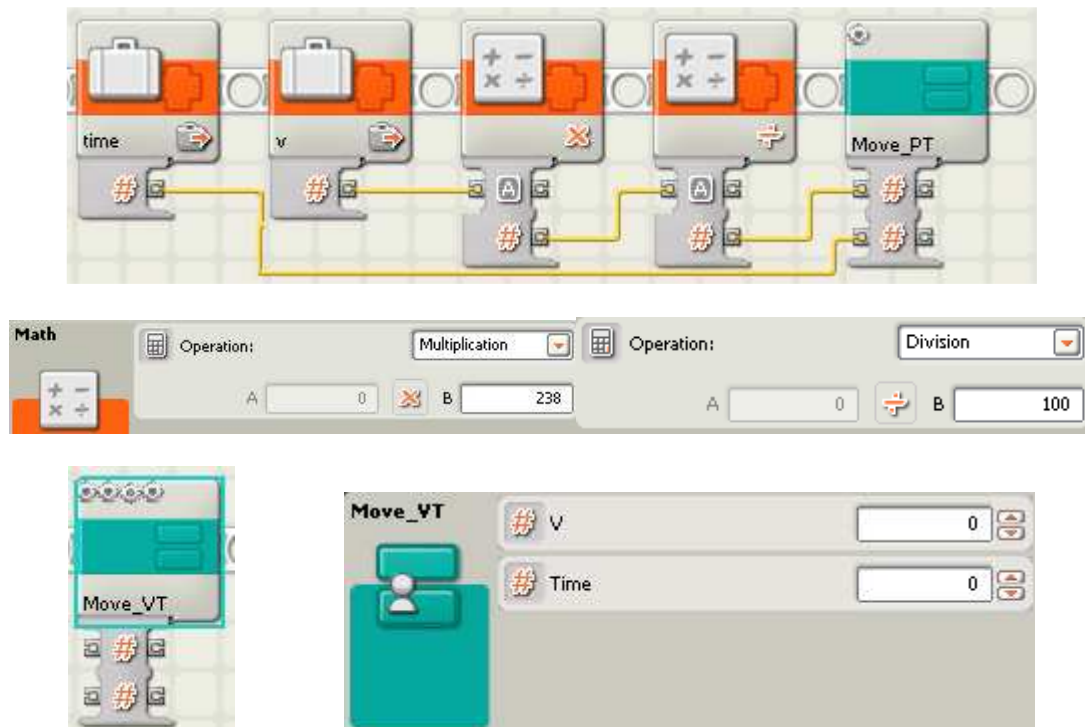
A partir de las exploraciones:

| POWER (%) | DISTANCE (cm) | TIME (s) | SPEED (m/sec) | THEORETICAL SPEED (=k * power) |
|-----------|---------------|----------|---------------|--------------------------------|
| 0 | 0 | 10 | 0 | 0 |
| 10 | 35 | 10 | 3,5 | 4,2 |
| 20 | 82 | 10 | 8,2 | 8,4 |
| 30 | 126 | 10 | 12,6 | 12,6 |
| 40 | 174 | 10 | 17,4 | 16,8 |
| 50 | 217 | 10 | 21,7 | 21 |
| k= | 0,42 | | | |



Observamos experimentalmente las distancias a la que se desplazan distintos robots con distintos valores de power durante un mismo tiempo, y calculamos sus velocidades. Generalizando estos casos vemos que la Operacion (v) -> power es $V = 0,42 * power$;

Lo que interesa es “alimentar” la entrada de Power del bloque Move_PT, asi que “re-escribimos” la formula como: $power = V/0,42$, o lo que es lo mismo $power = 2,38 * V$:



[Nota: Como el lenguaje NXT-G trabaja solo con enteros, multiplicamos por 238 y dividimos por 100]

4.- Conclusiones y perspectivas de transferencia

La robótica ofrece un alto interés por sus posibilidades educativas intrínsecas: permite desarrollar entornos de aprendizaje que facilitan la exploración de lo formal al estilo activo y constructivista en que Piaget trabajó lo concreto. Construyen por ello no sólo competencias específicas tecnológicas sobre el funcionamiento básico de los robots, sino competencias cognitivas generales sobre “aprender a aprender”, sobre “aprender por indagación” sobre “aprender a emprender proyectos y resolver problemas” y sobre el carácter esencialmente lingüístico del aprendizaje (competencias del catálogo de competencias básicas de secundaria).

Si queremos aplicar en el aula este tipo de tecnología debemos explorar los siguientes aspectos (además de otros como la formación del profesorado, los recursos disponibles, etc...):

- ¿Qué puede enseñar el profesor/qué pueden aprender los alumnos mediante actividades de robótica escolar?
- La robótica educativa ¿puede ser potencialmente un buen punto de encuentro de equipos de profesores de variadas competencias y disciplinas para promover
- buenas prácticas de aprendizaje constructivista en sus alumnos?
- ¿Cuales son y qué papel pueden tener las competiciones robóticas escolares?

Bibliografía

- [1] Piaget, J. (1972). The Principles of Genetic Epistemology. N. Y.: Basic Books.
- [2] Piaget, J. (1974), To understand is to invent. N.Y.: Basic Books.
- [3] Vygotski, L.S. (1978) Mind in society. The development of higher psychological process. Cambridge, Ma.: Harward University Press. Trad. Cast. de S. Furió: El desarrollo de los procesos Psicológicos superiores. Barcelona: Crítica, 1979.
- [4] Vygotsky, L. S.: Thought and language. [Mass] : M.I.T. Press (1968)
- [5] Ausubel, D. P.: Educational psychology: A cognitive view. New York: Holt, Rinehart and Winston (1968)

[6] POZO, J.I., Teorías cognitivas del aprendizaje, Ed. Morata, Madrid, 1989