# Bauhaus: Development of a walking robot

Francisco Padilla[1], Javier Porras[1], Israel Martínez[1], Luis Ávila[1], Carlos Soberanes[1], Juan Navarro[1], Jorge Barrera[1] and Leonardo Téllez[1]

[1] Universidad Panamericana
Augusto Rodin #498 Col. Insurgentes Mixcoac
C.P. 02920, México D.F.
agonzah@mx.up.mx

**Abstract.** Human being has been trying to create mechanisms that resemble living things all along his history in order to use the versatility they have. Within nature there are animals that develop a harmonic locomotion that allows them to overcome the obstacles in an apparently simple way. Walking machines try to get closer to these creatures' technological development, to have access to places where ordinary transportation would be very difficult.

Engineering students from the Universidad Panamericana, located in Mexico City, developed a walking robot that is capable to walk over any kind of surface.

## 1 Introduction

Flying machines as well as walking machines have a lot in common as both try to copy natural movements. As flying machines were seen by hundreds of people a few years ago, walking machines haven't caused much enthusiasm even today. Perhaps it's due to availability for millenniums, of the wheel. If there were a paved path between two points, vehicles with wheels would be faster than those with legs. Any way, there are many places over the earth where vehicles over wheels can't walk, approximately half of the earth, according to the U.S. Army [1].

Some advantages of walking machines:

- Obstacles.
- Can go up and down stairs.
- Can walk over irregular fields using a effective variation in the leg's length in order to equalize the surface undulations.

Investigations on walking robots have been focused on aspects that interrelate robot's control, the step analysis and the robot's mechanical design. These aspects are divided in two groups:

- Legs geometry.
- Performance and movement transmission power.

The way to lean of a walking machine is the figure of the polygon formed by the points where the foot of the robot touches the floor. More rigorously the support form is two points dimensionally placed in a plane that consists of a convex helmet of the vertical projection of all the points of the foot when this one is supported in the floor. So that a robot is statically stable, the vertical projection of the gravity centre must be within the support. Robots with static stability have at least three legs.

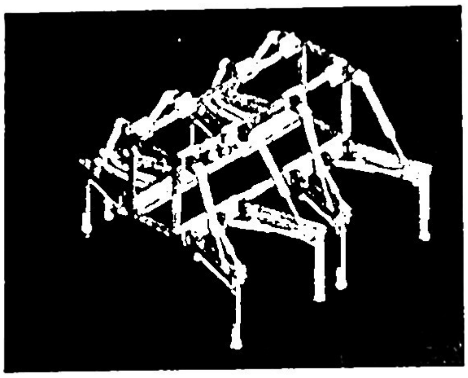

**Fig.1.** Bauhaus Robot

Project Bauhaus consists in build an octopod robot completely autonomous, mechanically animated with pneumatic cylinders that are sequentially activated by electronic valves. This robot is a direct evolution of Yolcatl and PeKe2:TB robots, built by our university in 2002 and 2003 respectively, this as a stronger development of the project in order to achieve a higher efficiency and use of the machine's resources.

Comparing with previous robots, we consider as primordial objectives:

- Weight reduction.
- Reduce air consumption.
- Simplify electronic circuits.
- Increase autonomy level.
- Simplify construction processes.
- Capability for future upgradeable

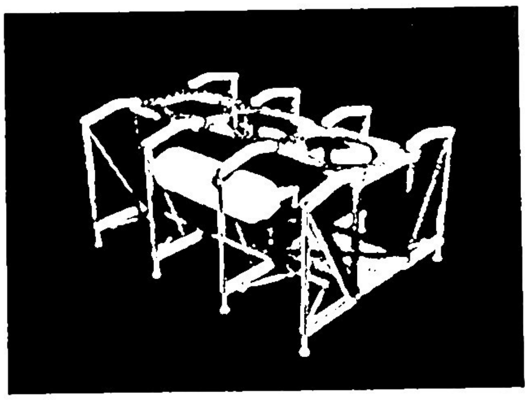

**Fig.2.** PeKe2:TB Robot

## 2. Mechanical Description

The Mechanical process design was established with the purpose to create a robust, fast and intelligent walking robot. In order to accomplish this objective, it was necessary to use several static and dynamic simulations, also a material analysis and a study of the finite element (F.E.M.).

The main objective in designing an insect-like robot is the fact that we want to obtain stability while walking in any type of surface. Nevertheless, it is difficult that a robot can obtain an absolute stability while walking through irregular surfaces, mainly if it is desired to have his legs in several heights.

### 2.1 Gait Analysis

One of the most important characteristics in this type of design is the versatility and the variety of gaits. Taking into account the step length, the stability and the air consumption, it was decided to have only one optimal walk type, although adjustments can be had bye changing the action sequence of the pneumatic actuators.

The robot's step length in each cycle depends on the angle the leg is moved, what depends as well of the horizontal actuators length move. A significant improvement from Yolcatl was that this robot has an adequate actuator position, managing to reduce the race of this cylinders and the air consumption by cycle.

Using the Working Model® software, it was simulated the robot movement, determining the most by this way the most adequate configuration. The configuration seeks to have a perfect symmetry between legs with angular equivalent movements, as much towards the front as backwards, with respect to the spin center of each leg, which is a fundamental affair if we want to have an optimal control mass area.

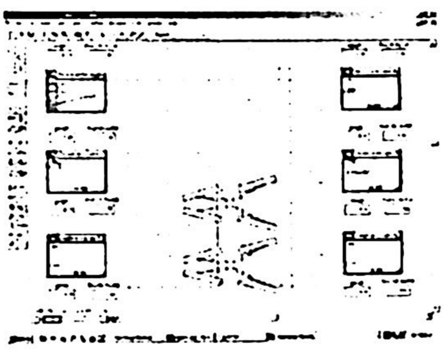

Fig.3. Working Model® display with Bauhaus' mechanical design

Nevertheless, given each leg configuration and the walking type, the robot has a little drag in the moment of driving the body forwards, which causes a little bit of inaccuracy in the total displacement. This point could be understood better if we solve the point on the ground as if they describe an overdeterminated four-legged mechanism that needs an additional degree of freedom. The problems carried by this drag may be reduced by the flux air regulation that gets through the leg's actuators, thus their consequences can be considered of little importance.

## 2.2 Lateral Scuff

The fact of walking using angular movements in the legs carries a problem of adjustment on the support position. When one leg is moved forwards and it is put at a certain distance without varying the spin radius, the distance between the leg support and the body is reduced. Now, if it is wanted to displace with one spin point in the new support, it is necessary a spin radius reduction. This change is adjusted by the passive degree of freedom and the ankles that are in the legs, being able the robot to walk without problems of deviations, excessive efforts in the legs or flexing momentums in the pistons rod.

## 2.3 Mass Distribution

In order to obtain a safe stability in a robot, three alternate strong points are necessary that distribute the weight properly. However, when a robot must walk through complicated surfaces, there is a risk of support lackness, which may cause the robot to fall. The fact that a single leg loads almost with half of the weight of the robot causes that the legs (at least the central ones) must be more robust and that the walk do not cause inertial forces that may be out of the movement axes of the robot. Nevertheless, the robot has movement in its mass center, which intermittently oscillates towards the right or the left but that when finishing the cycle of a complete step arrives at its starting point.

## 2.4 Turning System and Rotating Method

The robot has two rotating methods. The precision rotating system placed in the inferior part of the robot, has the purpose of making adjustments in the robot trajectory in case of unexpected deviations. A rotating actuator that permits 230° rotations, with torque of 10Nm, enough to beat the robot static inertia, composes the system. This Rotrik® is united to the robots body using bearings, to reduce the effects of friction. The base surface has been simulated, so that it would be the smallest possible, and to reduce the effects of inertia due to the robot's weight. In order to position the robot in the rotating system, it is not necessary to lay it down, but the robot sits down on it when the vertical actuators are closed. With this, the necessary height of the robot to rotate is minimal, having a major stability that enables us to rotate faster. The rotating actuator is in the base of the robot, not inside of it, so we are able to have more space inside the robot, also the base has more weight, gaining with this stability and reducing the falling possibilities.

On the other side, for turning purposes, four legs are raised and set each one at their corresponding position; left legs are set to the front while right legs are set to the back position. The other legs are set higher than the rest of them and external legs are moved to the opposite side. Several iterations of this method while varying cylinder displacement allow robot to rotate 360° clockwise (CW) or counterclockwise (CCW). A great advantage of this procedure is that robot rotates around an imaginary axis located at the center of body; this allowed robot to make turns in reduced areas.

## 2.5 Leg Design

Each leg has three degrees of freedom (up-down, left-right, front-back) that permit to have an optimal stability statically or in movement.

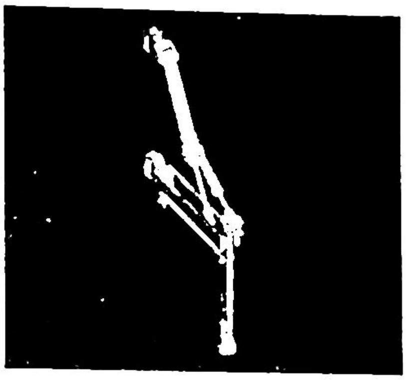

**Fig.4.** Bauhaus' leg mechanism

The horizontal portion on the leg is made of two parallel rectangular pieces of carbon fiber. A cold rolled iron round joints them and allows articulation with the rest of the leg and body. A detailed resistance analysis was performed in order to select material properties and specifications.

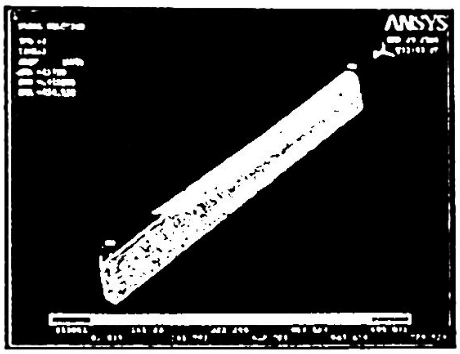


**Fig.5.** Horizontal bar stress simulation

The vertical leg portion was designed to raise the robot high enough to overcome obstacles that can be found along the track. They also allow the robot to walk at different heights.

By a set of parallelograms, legs are always maintained vertical and thus preventing robot to reduce its stability while ensuring that leg will support the same force anytime. However, when robot walks forward, vertical legs are pulled back due to angle variations at each step of walking. The variations mentioned above produce different sorts of stresses at inferior rounds of parallelogram. Those stresses created are absorbed by an iron spring, which also, allows leg to return to its perpendicular position against the floor when robot raises it back. Even that spring coefficient was calculated and tested, the leg is provided with two nuts and washers that allow adjustment of this system in case it has lost its elasticity and stiffness.

## 3. Pneumatic System

The pneumatic circuits provide to the robot the power needed to move him without an external power source, and give it the force to carry a bigger weight than its own. The system was kept very simple and compact, especially because of the AS valves which provide modularized support for pneumatic system.

Pneumatic circuit of robot consists of two independent sets of valves, both supplied by the same air tank. Each pneumatic circuit moves both horizontal and vertical cylinders, which are responsible of providing motion force to walking machine. Both circuits are controlled by an element connected in consistent series to a fast-escape valve activated directly by the emergency stop button. The reason of this fast escape valve is to fully free the air contained in system, except from air inside pneumatic cylinders. A third circuit provides the necessary air to power the circular actuator that permits the rotatory movement of the rotation system.

Finally, before air is sent to valves, a basic maintenance circuitry was added in order to eliminate possible humidity from air supply. This device is not required in spite of the air supply used. However, it was added for testing and maintenance situations in which air supply might be different.

## 4. Control System

Several factors were considered before choosing the controlling devices, including price, adaptability, performance, reliability and ease of use. Other important prerequisite was the multitasking feature, available on some controllers. Not only this allows getting better controlling but allows obtaining a smoother performance while walking. Although most single-board computers provide several built-in programming languages, just a few can be easily integrated without dealing with assembly-debugging or separated compiling. The time-critical support of the controller we selected, allowed us to mix assembly code and C code without precompiling and using far-calling conventions.

### 4.1 Programming Schemes

Programming system was based on a layered model similar to TCP, which provides reliable flow of data from entities. Also optimized code based on preprocessor usage reduces CPU overhead and allow mixed multitask to use spare cycles for making arithmetic calculations for next step. New features include a self-orientation system involving the auto leveler system.

### 4.1.1 Layered Programming Model

A layered model has several advantages against traditional programming model in which the code is just divided into functions that has no specific rules or that perform low and high level operations. Layered model is capable of being partially updated or upgraded while preserving its compatibility between each function. In addition, major modifications or improvements are fully supported by making just small modifications to concerning layer.

Another important advantage of this model is that as soon as a layer is operational and bug free, the rest of the code is built over this stable platform, ensuring that future development problems will be independent from lower level layers. Finally, layers make group-working easy cause each developer can be responsible of one layer without having to take in consideration lots of previously defined variable names or rules that involve the whole code.

Our model is divided in 4 layers as TCP is. OSI model involves a physical layer similarly structured as our model. The main reason of this is because our model was based on TCP. The layers used to design code were the following:

- Physical: Responsible of reading sensors and moving each cylinder to the indicated position.
- Walking and Positioning: Responsible of calculating robot position through various ways and detecting possible navigational errors and drifts.
- Path and Obstacle Identifier: Responsible of finding the shortest path and keeping track of obstacles or objects.
- Principal Task: Responsible of taking decisions and executing the different steps required to achieve the specific goal.

### 4.1.2 Mixed Multitasking

Microcontroller offered the possibility to implement both cooperative and preemptive multitasking without any serious problem or performance impact. Preemptive multitasking was used for fast or permanently updated tasks without releasing control to system or consuming full processor cycles for a period of time, for instance, for sensors. Cooperative multitasking provides a much more efficient and precise way of controlling high priority tasks or tasks that involves activating a specific mechanical device (in which most of the time is spent losing cycles).

Multitasking was established within four priorities to allow extremely important tasks to be running in foreground while slow tasks or low priority tasks are running in backgro and or whenever they are required for taking decisions. This scheme allowed the robot to be thinking while its moving reducing time before each decision is taken or then next step is initiated and while keeping code unmixed and modularized.

### 4.1.3 Extensive Pre-processor Usage

C Pre-processor was extensively used to reduce the number of functions and to reduce the size and number of parameters passed to functions. Also, the pre-processor was used to reduce code complexity while keeping its size small and high performance. At high level programming explicit directions like FRONT, LEFT, HIGH, or LOW are passed to functions making code readable and understandable. In the pre-processor, exactly before compiling, words are transformed into short integer constants that are passed much faster after compilation. Also, conditional compilation took an important part in developing code and testing process, allowing code to just include statements required at that time. The main function contains the multitasking code and robot initialization process, which is capable of setting the robot at home position and perform a quick self test. After that process the robot checks whether a specific trial was selected and prepares to execute the "Principal" task once. While the Principal Task is running a monitoring task called Sensors checks robot status and sends it to a global memory area, which was based on DMI concept. The whole set of tasks are being executed while the robot is trying to accomplish an specific duty, making it fully automated and creating a true "Act/React" system.

## 5. VISION SYSTEM

The vision system was created to fulfill the requirements of a versatile positioning system that could give us a real autonomy. The general idea was to recognize the lines and numbers on the stage in order to calculate the absolute position and direction over the stage. This recognition is made during 3 main stages, in the first we recognize the lines, next we search for a relative position and finally we get the absolute position.

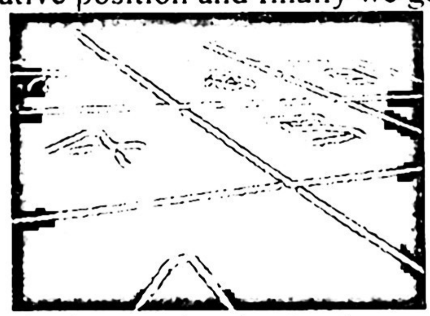

Fig.6. Track picture taken by Bauhaus camera

### 1$^{st}$ recognition phase

We start by taking a frame from the camera and then we process it by applying the Roberts Operator to enhance the edges from the image. We also calculate *a priori* some tables to optimize the next process, which is the Hough Transforms, this transform searches all over the bitmap using an accumulator array and lines represented in their $(\rho,\theta)$ form. Finally we throw away lines that can't form a perspective and then we calculate the intersection points between these lines.

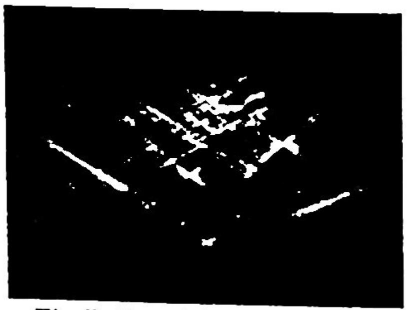

Fig.7. Hough Transforms

## 2nd recognition phase

The main idea behind this process is to apply an inverse rendering algorithm using the intersections points and the fact that the points that form the block we are looking for in the space must be equally distant and with 90° angles. The approach we take to solve this nonlinear problem was:

(a) Reduce the search range by using obvious facts about where the block we are looking for is.
(b) Scan each point on a line intersecting our range of search from beginning to end and looking if there is an optimal block that contains it.

After a successful search we made a coordinate system transformation, so we obtained the robot position relative to a block that is unknown by now.

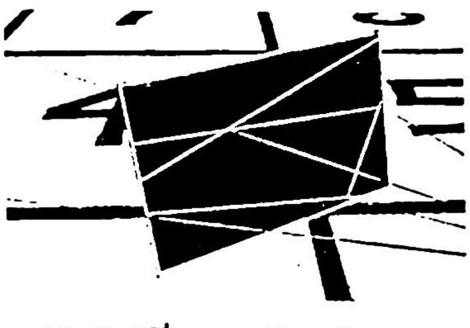

Fig.8. 2nd recognition phase

## 3rd recognition phase

The objective was to identify the block that the robot was seeing in order to transform the relative position into an absolute position. The approach we took was to compare the block identified with the mapping of a photo taken from each of the numbers. It was also needed to compute the mass center of each captured number to reduce the error from the Hough transform, then we assign a probability of being the correct number to each photo in the library and finally from the one with greatest probability we calculate the absolute position.

## 6. Conclusion

This robot presents the following advantages regarding its predecessor: Light weight; better stability during walking due to legs geometry; piston displacement reduced by 50%; air flow consumption was reduced by 40%; turning sequence optimized to a minimum amount of steps; there was developed a completely Re-design and improved vision system; programming code was reduced to its minimum.

Also we have many innovative approaches: modular design simplifies transportation and assembly; materials as carbon fiber and Epopast® used on robot's structure (Legs build with two different materials to enhance properties); and the use of finite element analysis simulation for validate the design of critical parts.

Then we have many troubles designing the robots that give us new ways of improvement and investigation: eight legged configuration generates a statically overdetermi-nated system, which leads toe small pulling in the legs; horizontal piston's position creates a certain waste in its race; horizontal piston's force is not equivalent in all race moments; there has to be certain angle compensation due to the angular movement of legs; used camera requires a great range of vision, because of that we need to put it in a high position respect the floor; and a great amount of light made difficult to distin-guish specifies objects.

## References


1. ANON (1967), Logistical Vehicle Off-Road Movility, Project TCCO 62-5, U.S. Army Transportation Developments Agency, Fort Eustis, Va.
2. BASUALDO, Raul, JACOBUS, Roberto; Designing a Walking Robot Through Computer Simulation, III Conference about Computer Simulation, Mexico City, 195pp 138-144.
3. Da Fortuna, L.; Shape Analysis and Clsification. Theory and Practice; CRC Press; 2001.
4. Watt A.; Advanced Animation and Rendering Techniques, Addison-Wesley; 1992
5. Glaeser, G.; Fast Algorithms for 3D-Graphics; Springer-Verlag; 1994