Development of a 3D laser scanner for guiding a six-legged walking robot

Samuel N Cubero, Benjamin C Frost
Department of Mechanical Engineering, Curtin University of Technology, Australia
s.cubero@curtin.edu.au

Abstract

This paper describes the mechanical design, control and performance testing of a 10mm resolution three-dimensional surface scanner with graphical display software, based on a 2D SICK^{TM} *LMS-200 laser scanner. This scanner will be used by the navigation system of a 6-legged walking vehicle (Hydrobug), currently being developed and tested at Curtin University of Technology, Australia. This robot was designed to walk over rough, broken ground or drive on four wheels over fairly smooth or flat terrain. Technical problems and future work planned for the development of a better 3D laser scanner for this walking vehicle are also described.*

1 Introduction

The "Hydrobug" robot ("Hydraulic Bug", the 6-legged, 4-wheeled robot depicted in Figure 1) is a 3-man passenger carrying, hydraulically powered walking vehicle that was designed by the first author since 1999 at Curtin University of Technology, Western Australia. At present, most of the mechanical hardware design has been completed, the hydraulic power pack has been built and one leg is operational under manual position control only.

One of the important goals for this walking vehicle is to travel in almost any direction that the driver commands by placing each of the feet automatically at safe positions on the ground or on hard supporting surfaces that are not too steep. The driver should be able to command the vehicle to drive forwards, backwards, sideways (left or right), rotate on the spot (clockwise or anticlockwise) and steer, travelling along a desired curved path (heading left or right). These types of commands can be issued via a video-game-type "Joystick", buttons on a PC keyboard or a custom designed controller, where the goal is to let the control software manage all the complexities of gait control and foot positioning automatically so that the human driver or operator can concentrate on the task of navigation and steering. This is a fairly straightforward problem to solve if the robot is travelling over a flat, level surface while maintaining a level orientation for its body. However, these types of movements are quite difficult to achieve over highly uneven surfaces or over unstructured terrain, as would be the case for example, if the robot is attempting to walk over large boulders near a seashore or over rocky surfaces on planet Mars. In order to achieve these types of walking motions automatically, the leg controller software needs an accurate geometric model of its nearby surrounding environment so that it can make the best and safest possible decisions for placing each foot. The robot must always remain statically and dynamically stable while making steady progress in the direction or movement intended. The top speed in walking mode is expected to be 5 km/hour so dynamic forces are not expected to be significant and do not need to be analysed. In wheeled mode, the top speed will be about 50 km/hour.

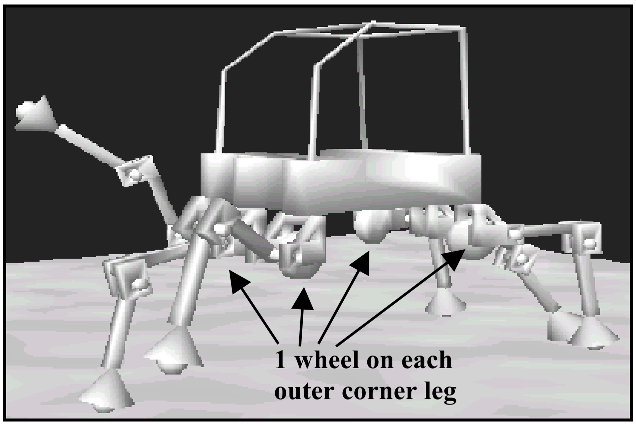


Fig 1. Hydrobug: A 6-legged, 4-wheeled robot

Large 6-legged walking vehicles, like the Plustech, Mechant and the DARPA ASV (Adaptive Suspension Vehicle) [1], were successfully able to walk over flat and moderately uneven undulating ground. However, these robots are very limited in being able to automatically walk over extremely irregular and rocky terrain involving large rocks, boulders, cliffs and deep pot-holes which are of the same order of size as their legs. Their controllers were virtually "blind" to the surrounding environment and could only react to or adapt to the ground that their feet came into contact with. Some walking robots, like the

Robug 2 (built during the 1980's), had to blindly search for suitable foot positions using pure guessing algorithms. This is clearly not the most efficient and fastest way for a walking robot to move safely and reliably over rough terrain. Details about these robots can be viewed on internet sites like the "Walking Machines Catalog" [1].

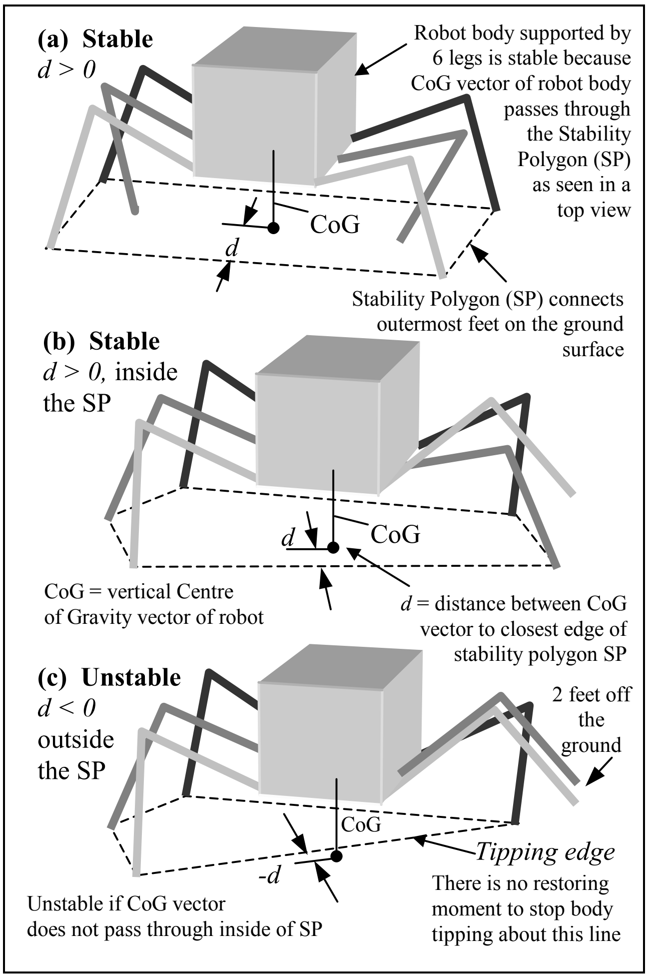


Fig 2. Stability Polygon for walking vehicles

2 Foot positioning and control

Software control algorithms can be written to ensure that the best possible foot positions are selected to maximize vehicle stability and movement performance. For example, each solid ground surface polygon, "patch", or ground position within the workspace of a robot foot can be assigned a "desirability factor" determined by its contribution towards maximizing the size of the robot's "Stability Polygon" (SP) and also on how flat the surface is, or how close the surface gradient is to zero relative to a horizontal plane. The flatness of a "patch", surface polygon or ground point is determined by the angle between its surface normal vector and the vertical "up" direction. The foot controller can select ground positions for advancing feet which have the highest possible "desirability factors" while continuing a regular gait pattern. To ensure stable walking, the robot's "Centre of Gravity" (CoG) vector should always pass through the interior of the "Stability Polygon" (SP), the stability margin ("*d*") should be kept as large as possible and surface gradients for new footholds must not be too steep.

Compliance for all legs must be controlled at all times so that internal static forces or moments on structural and actuator components do not result in excessive material stresses which could lead to mechanical failures. A real-time 3D kinematic model of the walking robot, joint angle data and data from 3 tilt sensors (accelerometers) on the robot body will allow the controller to accurately estimate the "(x, y)" position for the robot's vertical Centre of Gravity (CoG) vector, as seen in a top view, relative to a known point on the robot body. The robot's Centre of Gravity vector must pass through the "Stability Polygon", the polygon "SP" formed by the outermost supporting feet as seen in a top (plan) view, otherwise the robot will lose its static stability and fall over, assuming that the feet are quite small compared to the stability polygon's size. This fact can be confirmed using simple free body force equilibrium analysis on the entire robot body, taking into account the "top view" position of the CoG vector relative to the SP shape. Figure 2(c) shows two lifted front (right-side) legs which will create instability because none of the robot's legs can stop a "tipping moment" acting on the enire robot body about the "tipping edge" (the edge of the SP that is closest to the CoG vector). The distance "*d*", shown in Figure 2, is the shortest distance between the CoG vector of the robot and the closest edge of the Stability Polygon (SP). Many types of walking gaits may be used but the robot controller must maximize the value for "*d*" or always keep "*d*" positive at all times during gait execution, ensuring the vertical "Centre of Gravity" vector (CoG) always passes through the interior of the "Stability Polygon" (SP). Other factors that could cause instability or collapse of a walking robot include poor foot traction (related to surface steepness), which can lead to sliding, and breakage or buckling of one of the legs due to excessive loading. Hence, appropriate slip sensors and load cells (or force sensors) are necessary on the legs so that the controller can deal effectively with such problems.

3 Design of the 3D scanner

Topographical surface mapping is important for the walking robot controller to gain awareness of the shape of the surrounding ground surfaces as well as other important obstacles, steep cliffs, trenches or potholes which need to be avoided or traversed. A 3D model of the solid surface in front of the walking robot can be analysed in real time so that the control software can automatically select the best foot positions for maximising foot traction and walking stability while making progress over the terrain.

Various types of 3D surface scanners were considered, including vision systems employing "Structure From Motion" (SFM) methods, stereo (dual camera) vision, cameras that monitor bright stripes of reflected light for deducing 3D distances, radar and laser scanners. Each of these methods can be used for creating a 3D model of solid objects in front of a sensor and each method has its own advantages and disadvantages. For example, vision systems which rely on CCD 1D or 2D array optosensors require good contrasts (large differences in brightness levels) between foreground and background objects and minimum interference from shadows in order to perform reliably. This is not the best solution for outdoor applications, where shadows and direct sunlight can easily produce bad lighting and untrustworthy video images.

Probably the most reliable and versatile of these methods, even under undesirable or poor lighting conditions, is the "laser-range-finder" method of surface scanning. The distance to a solid object, which can reflect a strong "return beam" of laser light, can be calculated using the "time of flight" equation "Distance = Speed x Time", where "Speed" equals the speed of light (a laser beam) in air and "Time" is how long it takes for a laser beam to leave the scanner via a spinning mirror, reflect off an external solid object and return to an optosensor via the spinning mirror (ie. light is "timed" for the 2-way trip).

3.1 Basic operation of a SICK LMS scanner

At first, the author contemplated designing and building a 3D laser scanner from scratch, however, SICK^{TM} (Australia) kindly donated a LMS-200 2D laser scanner for experimental purposes. LMS is an acronym for "Laser Measurement System". The SICK scanner is able to measure radial distances to all solid objects around it by using a spinning mirror to sweep a laser beam across a flat 2D semi-circular area. It can transmit distance information (for each angular position of the mirror) to a PC in the form of bytes of serial data. The laser would be fired at a specific angle and the distance to a reflective object is measured and is transmitted back to the PC via a serial communications (eg. RS232 or RS422) interface. This process repeats at high speed, where measurements can be taken at every 0.25, 0.5 or 1 degree increments between a range of 0 to 180 degrees (see Figures 3 and 4).

The laser light is emitted in pulses from the range finder and travels to the object to be measured. The light is reflected back along the same path and is detected by the LMS-200. The time taken for the light to travel to the object and back is measured. The time measurement process involves a counter that starts when the light is transmitted and stops when the returned light is detected. This time data is converted to digital characters that represent distances from the scanner to the object being measured. In the case of the LMS-200, this digital character is dependent on the mode that the unit is operating in when the measurement takes place. The LMS-200 represents each single distance measurement with 13, 14 or 15 digital bits in a 16 bit string. The uppermost unused bits are used to represent other information that is also dependent on the mode of the LMS but could, for example, be used to represent the reflectivity of the surface being measured. The LMS also contains electronics that can measure the relative strength of the incoming signal and, thus, the amount of light reflected back from the surface being measured (almost like a CCD camera). This 16-bit data string containing the distance information for a particular scan is manipulated by the processing unit stationed within the LMS unit and, along with other scans, can be stored or sent to another peripheral unit such as a remote computer (or PC).

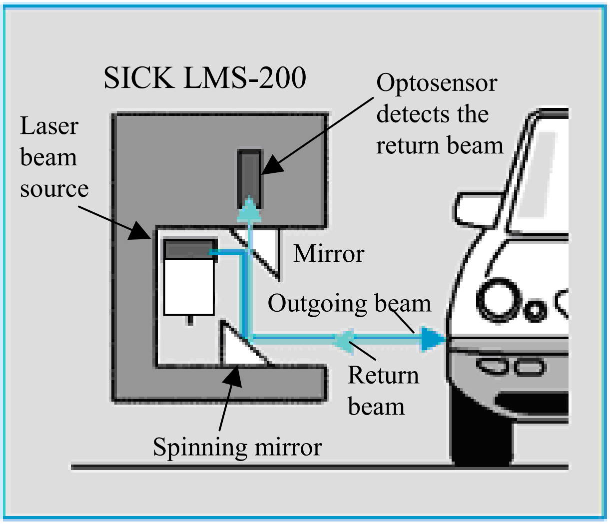


Fig 3. Spinning mirror (lower white triangle) reflects the outgoing beam and return beam [4]

The laser light is shot onto mirrored surfaces that are angled at 45 degrees to the horizontal (see Figure 3). The lower angled mirror rotates continuously, spinning in the horizontal plane in one direction so that the light from the laser beam source is swept in a circular manner in one plane. The laser light that is detected returns along almost the same path and passes through the upper mirror. The detection unit is stationed above the second unit. It is also noteworthy that the LMS can scan different sub regions, say, from 45 to 145 degrees. The scanner can also be set to scan different distances. It can be set to have a maximum distance scan of 8, 16 or 32 meters. Figure 4 shows a top view of the scanning range for the LMS.

The LMS-200's on board processor is pre-programmed to accept and execute certain commands from external sources and can adjust its own running modes and conditions depending on what it is commanded to do. Some examples of commands that can be sent are:

- Change the baud rate (from 9600 bps up to 500 kbps)
- Adjust the range of the scan (eg. Semi-circular 180 degrees or 100 degree scans can be selected)
- Change the resolution of scans (eg. Take a distance measurement every 1, 0.5 or 0.25 degrees)
- Change the measurement range (8m , 16m or 32m)
- Request a single or continuous scan (in mm or cm)

3.2 Serial communications with the LMS

The speed and format of the serial communication sent between the LMS and a PC must be compatible. Such settings include the common baud rate, number of data bits and inclusion or exclusion of stop bits. This information defines the format of each single byte of data that is sent to and from the LMS unit. For the LMS-200, the form of a byte is 1 start bit, 8 data bits, 1 stop bit and no "parity". The default baud rate is 9600 baud. The data that is transmitted conforms to the Intel^{TM} standard (Little Endian) which includes the protocol that when a data string consists of more than one byte, the least significant of those bytes is sent first. The format for commands are referred to as telegrams that include strings of 8-bit bytes or characters. (A byte is a string of 8-bits and a telegram is a string of bytes) Each telegram includes information to let the LMS know that a command of a certain length (in bytes) can be expected, which command form to expect and, finally, the command data itself. The telegrams are of the format shown in Table 1 (where STX stands for 'start of text' and "n" is the byte position, depending on the length of the data stream being sent).

Table 1. Form of an LMS telegram [4, SICK p.19] showing byte positions used in a telegram

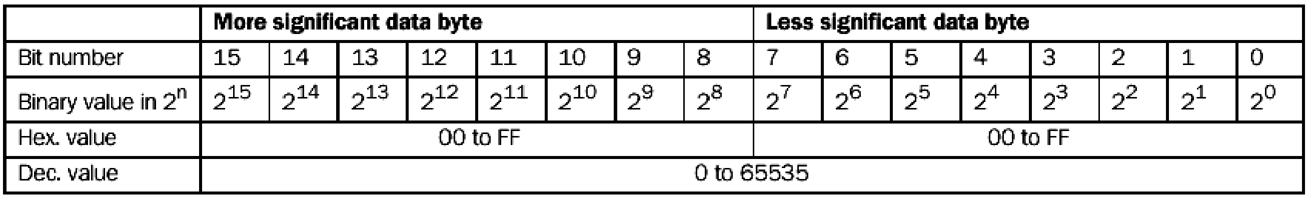
Frame		Commands & Data				Frame	
STX	Address	Length		command /response	data	Checksum	
1	2	3	4	5	6 to n	n+1	n+2

An example of a command telegram is that of the telegram used to change the baud rate. For example, to change the baud rate to 38400 bps, the PC must send the telegram: "02 00 02 00 20 40 50 08" (hexadecimal byte values to be sent in that order). Here the first byte "02h" (where the "h" represents hexadecimal) is the STX byte, which denotes the beginning of a telegram. The next byte "00h" is the address of the LMS then the following "02h" and "00h" represent the length of the command and data to be sent (note that this complies with the Little Endian format of least significant bit first). "20h" is the code for a baud change command and the command data itself "40h" is the choice of 38400 bps (bits per second) as a value for the baud rate. The last two bytes are the checksums. After this telegram is sent, the LMS outputs a confirmation or a reply telegram (in hexadecimal number format) "06 02 81 03 00 A0 00 10 36 1A" back to the user (ie. the PC). Every time a command (or telegram) is sent to the LMS, a reply telegram is sent back to let the user know that the command was received and carried out.

The reply telegram is of an identical format to the sent telegram except for the fact that it includes an extra bit before the (STX) "02h" byte that represents the beginning of a telegram. This extra bit, "06h", sent before the telegram's official beginning, is the acknowledge byte and acknowledges that a telegram was received by the LMS unit. It is important to note that the address byte is different because the address of the LMS and the address of the PC used to communicate with the LMS are different. The address of the LMS is "00h" and the address of the PC, in the case of the examples above, is "81h". A complete list of the LMS serial communications protocol, describing all command telegrams and reply telegrams, is published in the document by SICK AG. 2003: "Telegrams for Operating/Configuring the LMS 2xx" (Firmware V2.10/X1.14) [4].

Once the operating modes of the LMS are set to their correct values then scans can begin. The settings for a scan, such as start angle, finishing angle and the resolution of each scan taken (in 1°, 0.5° or 0.25° angle increments for the spinning mirror), must be set before the scan begins. The LMS user can send a telegram requesting a single scan to obtain a 2D "snapshot" of radial distances to solid light-reflecting surfaces of objects lying in the same horizontal plane as the LMS.

Table 2. 16-bit coded form of output distance (data chunk) sent by the SICK LMS-200 [4, SICK p. 17]



	More significant data byte								**Less significant data byte**							
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Binary value in 2^n	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Hex. value	00 to FF								00 to FF							
Dec. value	0 to 65535															

For example, the LMS can scan a full 180 degree range for the spinning mirror (as shown in Figure 4) at 1 degree resolution, or it can perform continuous scanning for "real time" measurement updates. However, in either case, the coding for each individual scan is the same. The distance information for a single scan is returned to the PC in the form of a return telegram wherein the distance data is coded in 16-bit chunks in the data section of that telegram. (See Table 2) The telegram requesting a single scan is command "30h". [4, SICK AG, p. 44]

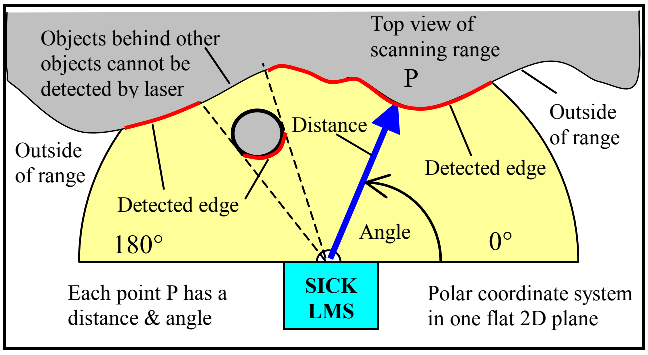


Fig 4. Scanning range for SICK LMS-200

Depending on the mode of the LMS, 13, 14 or 15 of the least significant bits of the returned code are used to represent the distance data. (Accurate to about ±1 units)

- 13 bits for the 8m range mode (0-8191 or 2^{13} values)
- 14 bits for the 16m range mode (0-16383 or 2^{14} values)
- 15 bits for the 32m range mode (0-32767 or 2^{15} values)

The rest of the bits can be used to represent reflectivity or can be ignored altogether. The data is coded as direct mm or cm data. For example, if the LMS is set to "mm" mode for an 8m scanning range, and the 13 bits of the data chunk is equivalent to the decimal value 700, then the actual measured distance is 700mm for that one reading.

3.3 Mounting theSICK LMS for 3D scanning

The SICK LMS-200 2D scanner alone is not enough for capturing 3D surface data in a stationary position, as it needs to be passed directly over the terrain in order to obtain new sections to scan. To scan 3D surfaces, the 2D scanner can be mounted on a tilting unit (rotating shaft axis), driven by a position controlled motor, controlled by software. A tilting unit, like that shown in Figure 6, can sweep the 2D laser scanner over a defined range of angles in a pitching type motion to measure 2D radial distances at each tilt angle. This method can produce accurate measurements to many grid points on solid objects within a 32m distance from the sensor. These radial distances were transformed and plotted on a computer screen in the form of nodes on a 3D surface mesh. At present, each 3D scan for an entire scene takes about 35-45 seconds to capture, depending on the baud rate used (this will be discussed in a later section). This time depends largely on the size of the angular range of the tilting platform, the baud rate for communications with the LMS-200 scanner and the efficiency of the software used to generate the 3D graphics. Theoretically, it is possible to capture a high-resolution 3D scene "snapshot" at up to 0.37 frame per second using a proprietary SICK PC communications card. For high-speed 3D scene scanning, higher framerates would be desirable for monitoring moving objects and a fast changing environment (eg. scanning moving cars, pedestrians or other obstacles which could collide with the robot.) High framerate 3D laser scanners are also ideal sensors for real-time monitoring and warning of dangerous driving or potential collisions with other objects, especially for motor vehicles. Weingarten, Greuner and Siegwart [2] describe a 3D scanner based on the design of a SICK LMS 2D scanner. The "Groundhog robot", built by Carnegie Mellon University, uses a similar method of 3D scanning for mapping mine shafts.

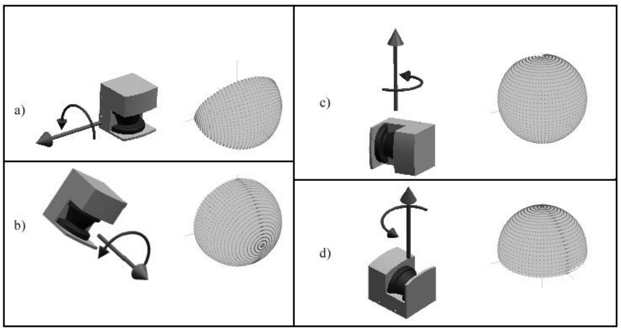


Fig 5. Scanning a 3D volume [3, Wulf & Wagner]

For a walking or mobile robot, important surface features that the leg controlling software must analyse include surface slope and reachability. Each surface "patch" (or rectangular surface bounded by 4 node points on a wireframe mesh), represents a solid surface and can be analysed in real time and independently assessed for suitability as a possible position for foot placement. When using a spherical "ball-shaped" foot on each leg, the less steep or flatter the contact surface is, the better the traction or friction force will be. Surface gradient is measured by the angle between the surface patch's normal vector and the upwards vertical direction. If the ground slope (or gradient of the surface patch) is too steep relative to the horizontal plane, this area is unsuitable for obtaining a good foothold because the foot could slip or slide off the surface due to insufficient friction. If a scanned ground position is too far away for a foot to reach it, it can be easily avoided and a better foothold position may be found that guarantees the largest possible stability margin ("*d*") while continuing the current gait pattern.

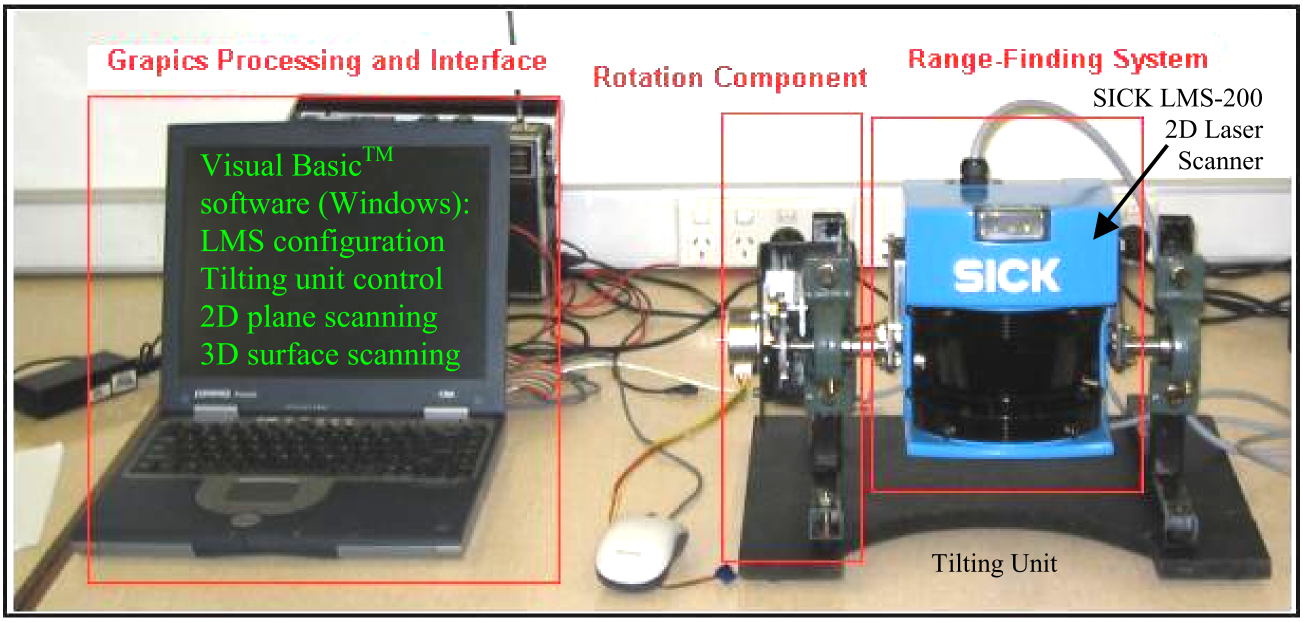


Fig 6. SICK LMS-200 2D laser scanner mounted on a custom built tilting unit controlled by a laptop PC

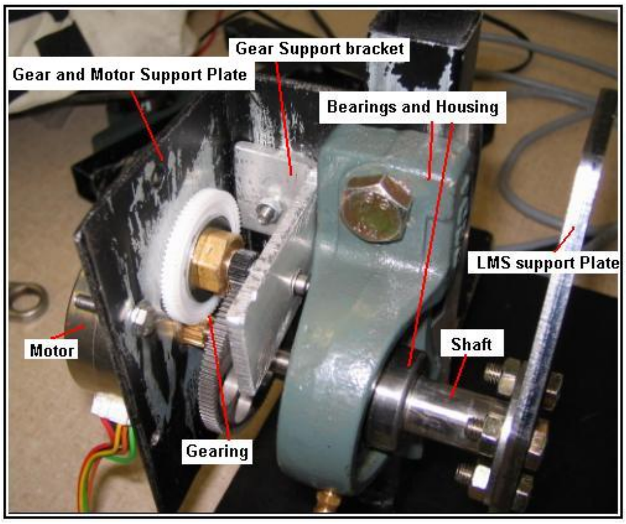
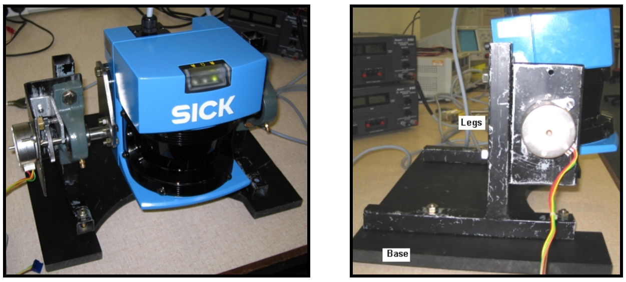


Fig 7. Stepper motor & gearing for tilting unit



(a) (b)

Fig 8. Tilting unit for pitch rotation of SICK LMS

Figures 6 and 7 show some mechanical components for the tilting unit. An off-the-shelf unipolar stepper motor, driven by MOSFETs (connected to a standard PC parallel port) is used to set the pitch angle for the LMS-200 2D laser scanner, which is mounted on two metal support plates. To minimize friction and wear on components, ball bearing units are used to support the tilting shaft. Figure 8(b) shows a side view of the tilting unit.

4 Controlling the 3D scanner hardware

The 3D scanner requires two units to be controlled by software running on a Windows^{TM} PC, namely: (1) the custom designed and built tilting unit in Figure 8(a); and (2) the actual SICK LMS-200 2D scanner mounted on the tilting unit. The control of these units is described next.

4.1 Electronic hardware for the tilting unit

The tilting unit is driven by unipolar stepper motor. The motor chosen to drive the system was a 7.5º Stepper Motor (Minebea^{TM} PM55L-048). It was salvaged from an HP^{TM} DeskJet^{TM} 500C printer. The actual method for calculating the degrees for each step was achieved through counting how many steps it would take to make the LMS rotate 720 degrees. The number of steps taken was 2301 and thus a single step was equal to 0.3129º of LMS rotation. This corresponds to the ratio 23.96:1 for the gear ratio, where $7.5^{o} \div 23.96 = 0.3129^{o}$ for a single step.

The stepper motor is controlled from the parallel port of the computer. The circuit designed to control the stepper motor essentially involves 4 MOSFET switches being switched "on" and "off" by the four least significant bits of a PC parallel port (LPT1). These devices are shown in the schematic of Figure 9. The "buffer unit" is a 74HC244 octal buffer/line driver, which holds onto the value of the lower nybble outputted by the Printer Port. The MOSFETs are MTP3055VL logic level switches.

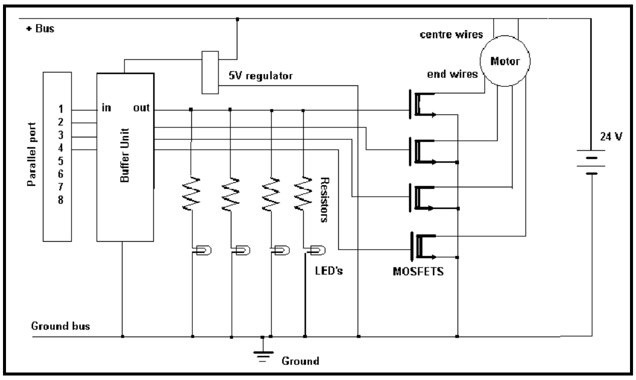


Fig 9. Stepper motor driver circuit driven by PC

Bits 1, 2, 3 and 4, as shown in Figure 9, are set to specific bit patterns in a particular sequence in order to control the forward or reverse stepping movements of the motor. Figure 8 shows how each of the four coils of a stepper motor (ϕ_1 , ϕ_2 , ϕ_3 , ϕ_4) , each driven by one MOSFET switch, can be energized (ON) or de-energized (OFF) in order to rotate the motor clockwise (CW) or counter-clockwise (CCW), for the full-step and half-step methods of stepping. For example, to rotate CW in full-stepping mode, output Steps 1, 2, 3, 4, 1, 2, 3, etc. For CCW rotation, output Steps 4, 3, 2, 1, 4, 3, 2, 1, etc. Greater precision can be obtained using "half-step" phase stepping sequences applied in CW or CCW order.

Unipolar full-step phase sequence

	Step	ϕ_1	ϕ_2	ϕ_3	ϕ_4
CW	1	ON	OFF	ON	OFF
↓	2	ON	OFF	OFF	ON
CCW	3	OFF	ON	OFF	ON
↑	4	OFF	ON	ON	OFF

(*Note:* CW = Clockwise; CCW = counter-clockwise)

Unipolar half-step phase sequence

	Step	ϕ_1	ϕ_2	ϕ_3	ϕ_4
CW	1	ON	OFF	ON	OFF
↓	1.5	ON	OFF	OFF	OFF
	2	ON	OFF	OFF	ON
	2.5	OFF	OFF	OFF	ON
CCW	3	OFF	ON	OFF	ON
↑	3.5	OFF	ON	OFF	OFF
	4	OFF	ON	ON	OFF
	4.5	OFF	OFF	ON	OFF

Fig 10. Stepper motor phase sequence examples

4.2 Control software implementation

In this section the design of the software system is explained briefly to show how it controls the LMS and the mechanical tilting unit. The SICK LMS-200 unit that the authors experimented with had a built-in RS232 serial port which could connect to a standard PC COM1 serial port or to a 500 kbps SICK card using RS422. Some of the commands that can be sent to the SICK LMS-200 using the SICK software (for Microsoft Windows^{TM}), include:

- Change the Baud rate of the LMS
- Change the resolution and spinning mirror scan range
- Request a single 2D scan and show the results as a graphical plot of (radial) distances of solid surfaces from the scanner (1 scan is a single 2D "snapshot").
- Request continuous scans with constant updates of the graphical plots in real-time
- Change the angular scanning range for the LMS

After becoming familiar with the main functions of the LMS scanner using the proprietary SICK software, the authors decided to develop their own scanning software using the communications protocol described in Section 3.2 in order to execute all of the following operations:

1. Set the minimum and maximum angular range and incremental angle for the tilting unit and perform all of the above configuration/initialisation settings for the SICK LMS (this is done manually by the user)
2. Tilt the LMS to the lowest (minimum) tilt angle
3. Request a single 2D scan for the current tilt angle
4. Receive and store measured distances for all points at the current tilt angle (each point has a distance, mirror angle, and tilt angle; ie. 3 spherical-polar coordinates)
5. Increment (raise) the tilt angle by a small angle and return to Step 3 until the maximum tilt angle has been scanned and data for the entire 3D scene is captured
6. Plot the graphical representations of scanned surfaces using 3D wireframe mesh graphics in Visual Basic^{TM}.

The final step above will not be described here for the sake of brevity, however, it involves converting each spherical-polar coordinate of a measured point (eg. Point P in Figure 4) to {x, y, z} cartesian coordinates and transforming 3D points to 2D points for point-to-point line plotting in a viewport window. This requires extensive use of homogenous transformation matrices [5]. Each of the lines plotted forms the edges of neighbouring surface patches (polygons) which, when joined together, form a surface "mesh" or wireframe model of the entire surface. The 3D scanner control software was written entirely in Visual Basic^{TM} 6.0 for Windows^{TM}. It allows the user to set horizontal and vertical scope (minimum and maximum angular ranges) and the horizontal and vertical resolutions for taking distance measurements. The control software also allows manipulation of the plotted graphics so that 2D and 3D scans can be viewed from different angles.

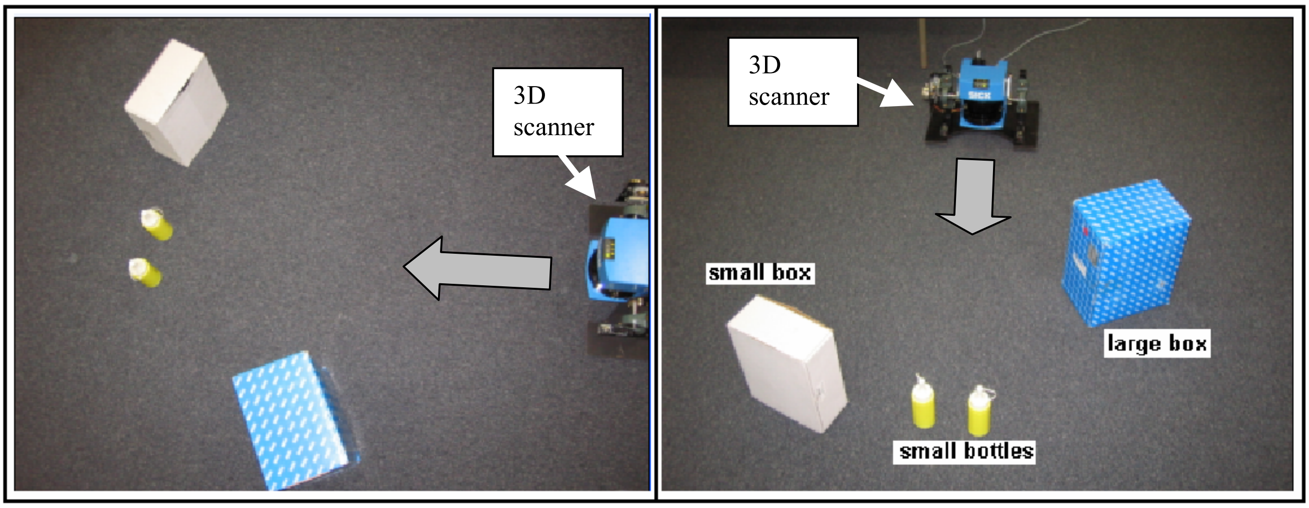
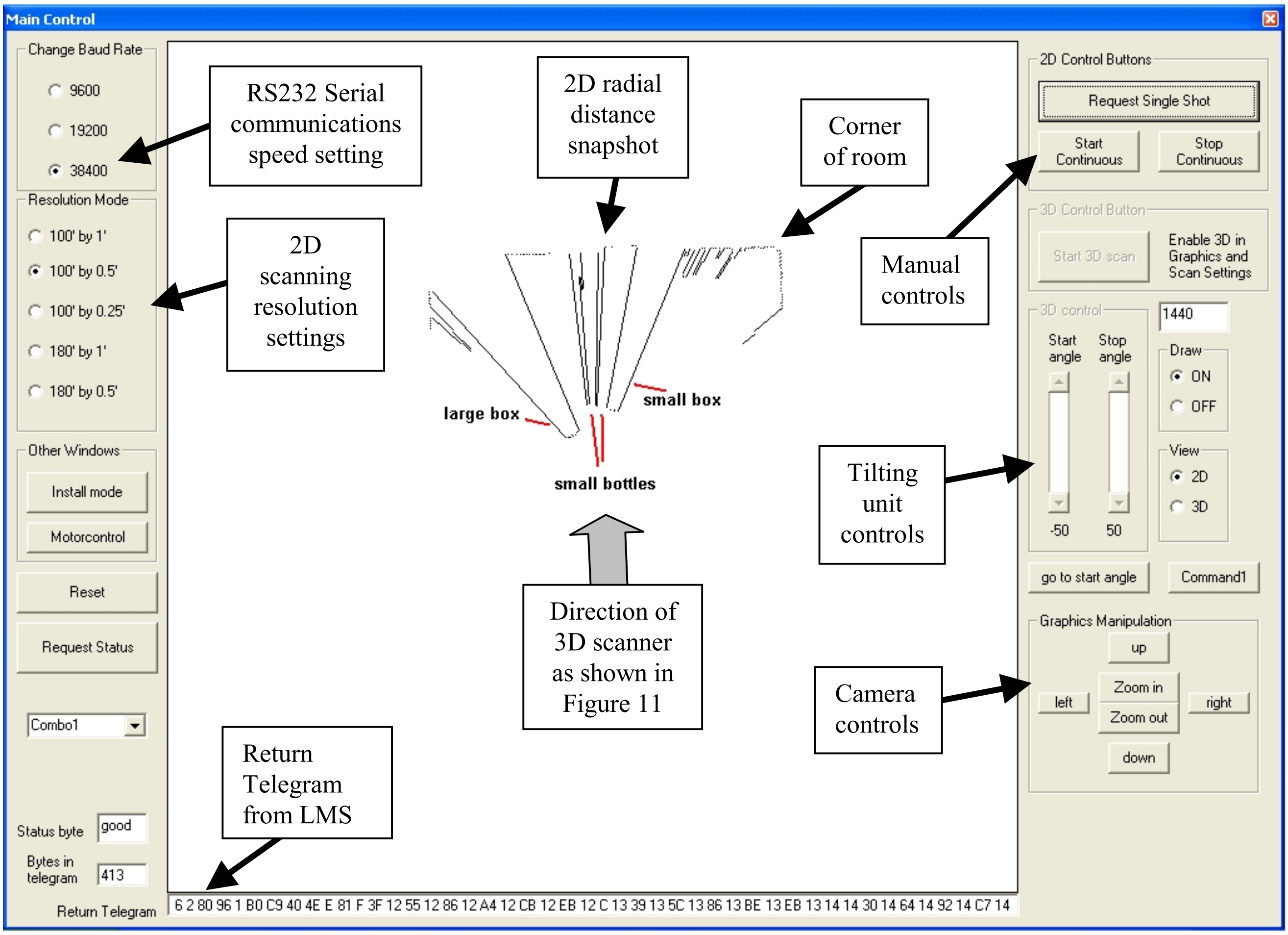


Fig 11. Example scene for scanning (results in Fig 12)



Change Baud Rate
○ 9600
○ 19200
● 38400

Resolution Mode
○ 100' by 1'
● 100' by 0.5'
○ 100' by 0.25'
○ 180' by 1'
○ 180' by 0.5'

1440

Draw
● ON
○ OFF

View
● 2D
○ 3D

Status byte: good

Bytes in telegram: 413

Return Telegram

```
6 2 80 96 1 B0 C9 40 4E E 81 F 3F 12 55 12 86 12 A4 12 CB 12 EB 12 C 13 39 13 5C 13 86 13 BE 13 EB 13 14 14 30 14 64 14 92 14 C7 14
```

RS232 Serial communications speed setting

2D scanning resolution settings

2D radial distance snapshot

Corner of room

Manual controls

Tilting unit controls

Direction of 3D scanner as shown in Figure 11

Camera controls

Return Telegram from LMS

Fig 12. Graphical User Interface showing a 2D “snapshot” of radial distance data for objects in Fig 11

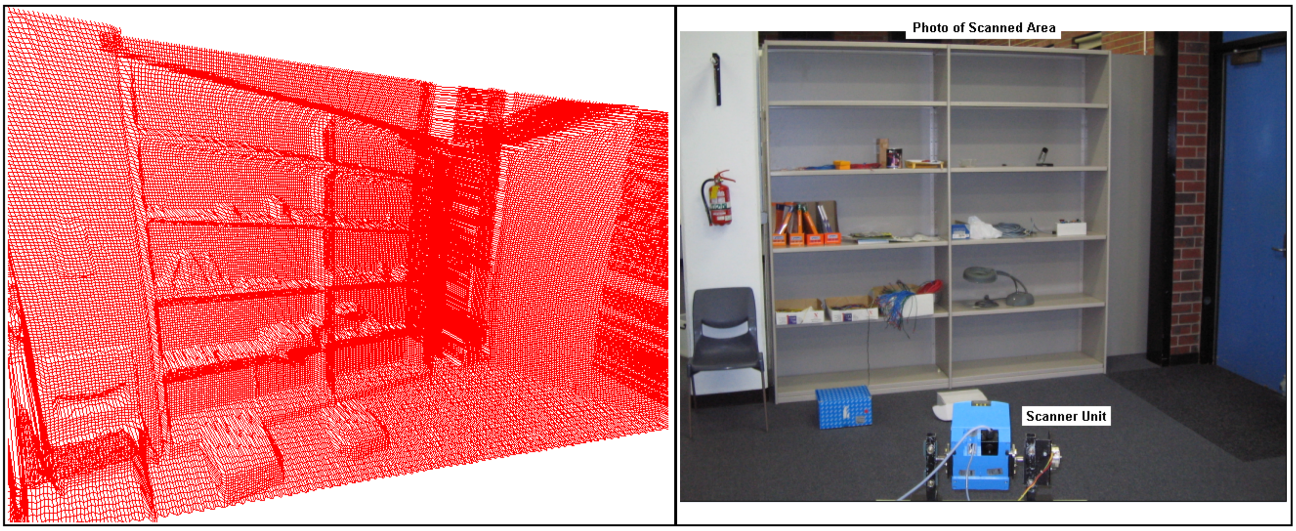


Fig 13. 3D scan of a bookshelf at the Mechatronics Studio (Curtin University of Technology, Perth)

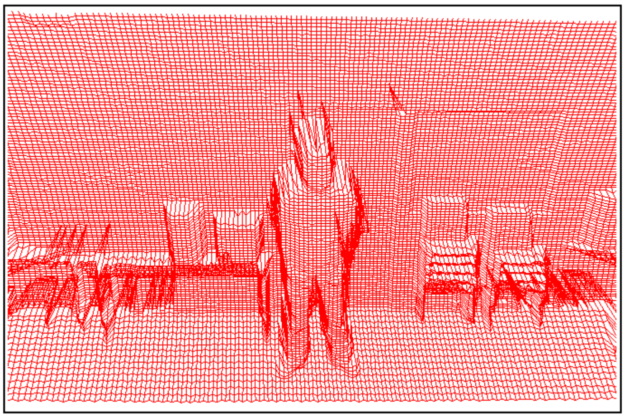


Fig 14. 3D scan of person standing in a room

5 Scanning results

Figure 12 shows the Windows^{TM} GUI (Graphical User Interface) for the 3D scanner control software. In this same screenshot is a "2D radial distance plot" of the reflective edges of solid objects that were found within the SICK LMS scanner's sensing range. (Compare Figure 11)

Figures 13 and 14 show some more examples of high resolution scans of 3D surfaces. Each of these scans took about 35 seconds to capture and 2 seconds to plot using a 38400 baud rate but speed depends heavily on the total number of 3D points captured. Once this surface data is available to a mobile robot controller, object detection algorithms can be used and gradients (steepness of normal vectors) can be calculated for each "patch" (rectangular polygon) over the entire 3D scene, or over a small area of interest where steep gradients or objects are found.

Fig 15. Prototype leg & wheel of the Hydrobug

Fig 16. Front view of Hydrobug body showing mounted hydraulic 'powerpack' & prototype leg

6 Observations and future work

Some obvious problems that were noticed during the development of this project, include:

- Very shiny, mirror-like surfaces tend to produce "infinite", very large or erroneous radial distance measurements which needed to be filtered out or ignored. (eg. very reflective paint on doors)
- The fairly weak torque output for the stepper motor could easily cause slipping of the tilting unit in the event the entire unit is jolted or bumped in the vertical direction. A closed-loop (eg. PID position controlled) tilting unit, with position feedback, could improve accuracy and resistance to such disturbances.
- The time to capture a detailed 3D scene is unacceptably long (35 seconds), rendering this 3D scanner unsuitable for guiding fast moving mobile robots and AGVs (Automatic Guided Vehicles). A "move a little, stop, capture scene, move a little, stop, capture scene" sequence would need to be repeated in order to avoid collisions and navigation problems.
- The maximum serial communications speed, using RS422 and the SICK data capture card is around 500 kbps, about 13 times faster than the 38400 bps baud rate through the PC COM1 port, so theoretically, the fastest 3D capture time that can be achieved with a SICK LMS-200 would be around 2.69 seconds, or a 3D scene capture framerate of 0.37 frames per second. This is, unfortunately, still quite slow for a mobile robot controller that needs to react responsively to a changing environment, especially when travelling at high speeds (eg. in driving mode).

The authors hope that this paper will be helpful to those wishing to develop their own 3D scanner using a 2D LMS scanner. The LMS-200 laser scanner that was tested by the authors retailed for about AUD $8,500 (Australian dollars) in 2004.

The Hydrobug robot may need to be fitted with at least 4 LMS-style scanners (facing the front, back, left and right directions) to scan all surrounding ground surfaces.

Figures 15 and 16 show photos of an articulated, hydraulically actuated leg and wheel suspended from one corner of the main body of the first Hydrobug prototype robot. This robot leg and the hydraulic powerpack are currently operational and can be remote controlled by a PC. Future work includes making improvements to the hydraulic power pack, installing more instrumentation and onboard sensors to accurately monitor operating variables, establishing two-way, long range, high-datarate RF communications to a "base station" computer and incorporating GPS (Global Positioning Satellite) navigation to work alongside several 3D scanners, a multi-camera vision system and other important sensors (eg. inertial sensors, accelerometers, tilt sensors, force/load/ slip/contact sensors on the legs, etc.) It is expected that the total materials and parts costs to complete the entire robot will come to around AUD $100,000 (in Australian dollars, 2005). The largest component of this total cost will probably be the 3D laser scanners, therefore, it is important to bring this cost down as low as possible, perhaps by redesigning and building better 3D scanners.

7 Limitations of commercial 3D scanners

There are several 3D scanners currently available on the market which perform a similar function to the custom built 3D scanner described in this paper. The design of the Callidus^{TM} [6] 3D scanner is also based on the SICK LMS and is used for scanning building interiors for architechtural and dimensional measurement applications. Unfortunately, this system is just as slow at capturing the same number of points as the system built by the authors as described in this paper. FARO Technologies^{TM}, which recently acquired iQvolution^{TM}, markets the FARO^{TM} LS 880 HE40 (indoor), HE80 (outdoor) and iQSun^{TM} 880 3D laser scanners. [7] These systems are reportedly able to capture an almost spherical 3D scene (360° field of view in the horizontal, 320° in the vertical) in a time of about 160 seconds, to an accuracy of 3mm error. The iQSun 880 measures 15.7in x 6.3in x 11in (inches) in size, however, these systems capture measurements at very low speeds making them very unsuitable for high-speed applications requiring quick responsiveness by a controller. Perhaps the fastest surface scanner on the market, at the date of this writing, is the SICK IVP Ranger^{TM} M50 vision-based 2D scanner [8] which uses a patented CMOS sensor consisting of 1536 x 512 pixels, 1536 A/D converters and 1536 parallel processors all in one IC package. A laser beam is used to form a straight line or a strip (plane) of light which is projected onto a solid surface and the laser light is analysed in the video image to judge distance of points from the camera. The SICK IVP is reportedly capable of scanning up to 10,000 profiles (line scans) per second, or up to 15 million measurements per second. Unfortunately, the IVP is currently only a 2D (plane) scanner and must be mounted on a tilting or sweeping platform in order to capture a 3D scene from one location, similar to the SICK LMS.

As mentioned earlier, vision-based measurement systems perform their best under controlled lighting conditions, especially in indoor environments. They can be susceptible to measurement errors if the surface or camera is exposed to strong sunlight, as in the case of outdoor situations. Even laser scanners can produce erroneous results when their sensors are pointed directly into the sun. Since this is rarely encountered in outdoor applications, laser-only based measurement systems like the SICK LMS tend to be more robust and reliable than vision systems in outdoor, sunlit environments, except for surfaces which happen to be highly reflective (eg. water, glossy paint, etc.) or which cannot return sufficient reflected laser light. Hence, this is why laser scanners are still the number one preferred choice for long-distance, high-precision distance measurement and 3D scanning. Unfortunately, most, if not all, commercial 3D scanners suffer the problems of being too large, heavy and costly.

8 Suggested improvements for 3D scanners

Considering all of the above limitations of commercial 3D scanners, it is clear that there is a real need for better performing and better value 3D surface scanners that are:

1. faster, producing measurement speeds as good as, or better than, the SICK IVP scanner. Conventional 3D laser-only scanners are unable to capture highly detailed 3D scenes at high framerates (eg. over 25 FPS)
2. smaller, compact and lightweight enough for use on small mobile robots and even aerial or VTOL flying robots. (eg. "International Aerial Robotics Contest")
3. low cost and affordable (eg. under AUD\$1000), to make it easily accessible to Universities and students.
4. simpler and easier to manufacture, with fewer moving parts and mechanisms, which would contribute to better reliability, less wear and lower production costs.

9 Conclusion

This paper described the most important theory for automatically positioning the feet of a 6-legged walking vehicle using information provided by a 3D laser scanner. The design and control of the mechanical tilting unit and the SICK LMS-200 2D laser scanner was combined and implemented to produce an operational 3D laser scanner with a 3D graphics plotting system and a user friendly Windows^{TM} PC interface. This 3D scanner can be used for many different types of measurement applications, however, it was concluded that the slow capture speed of this system makes it unsuitable for controlling the Hydrobug robot at high speeds, especially in driving mode. A faster and lower cost solution is needed.

SICK have expressed interest in supporting research to develop high-framerate, low-cost and lightweight 3D laser scanners to meet the portability and speed requirements of the Hydrobug project and aerial or VTOL flying robots in general. Future work is currently being planned to develop the next generation of 3D scanners which are expected to be very small in size and cost while being able to deliver high-speed 3D scene capturing framerates in excess of 25 frames per second for fast moving mobile robots and road vehicle safety and monitoring applications. For example, a 3D scanner can be mounted on a car or a truck to detect nearby activity or objects around it. An onboard computer, continually analysing 3D distance data, can trigger an "early warning" alarm or notify the driver of a potential impending collision with a nearby vehicle or object. (eg. a vehicle in a neighbouring lane, other cars, a tree, a sign post, the road shoulders or edges, gutters, pedestrians, cyclists, etc.) If the driver falls asleep at the steering wheel, such a system could wake up or warn the driver that his or her vehicle is drifting out of its marked road lane and is driving dangerously, especially if the drifting behaviour occurs without an indicator light or turn signal being active. If you happen to forget to take a quick look at your "blind spot" while driving and you begin to make a lane change which may side-swipe another vehicle in a neighbouring lane, this type of safety system may be able to warn you in advance about the presence of vehicles that are too close or moving towards you too fast, thus, vastly improving driver awareness.

It is hoped that the Hydrobug will be tested for walking over very large boulders, extremely broken ground and very steep surfaces within a few years time. With enough of the right kinds of sensors, the Hydrobug's control software can be made highly aware of its environment and its own mechanical condition (eg. foot positions, CoG vector position, etc.), thus, enabling the robot to carefully select the best possible footholds or ground support points; a task that fixed-axle, wheeled vehicles cannot accomplish mechanically. Walking robots like the Hydrobug could be useful for remote controlled or semi-automated construction, farming, surveying and all types of tool manipulation work, on land, under water or even on other planets or space stations.

References

1. CLAWAR: Walking Machines Catalog
   URL: http://www.walking-machines.org
2. Weingarten, Greuner and Siegwart, 2004: A state-of-the-art 3D Sensor for Robot Navigation, Swiss Federal Institute of Technology and Swiss Centre for Electronics and Microtechnology (CSEM), Switzerland
3. Wulf & Wagner, 2003: Fast Scanning Methods for Laser Measurement Systems, Institute for Systems Engineering, University of Hanover, Germany
4. SICK AG. 2003: Telegrams for Operating/ Configuring the LMS 2xx (Firmware V2.10/X1.14), www.sick.com , Germany.
5. Rod Stephens, 1997: Visual Basic Graphics Programming, John Wiley and Sons, Inc, New York. ISBN: 0-471-15533-0
6. TRIMBLE www.trimble.com/callidus.html
7. FARO and iQvolution: www.iqvolution.com and www.manufacturingtalk.com/news/iqv/iqv101.html
8. SICK IVP www.sickivp.se & www.chronos-vision.de