(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 January 2018 (18.01.2018)

WIPO | PCT

(10) International Publication Number
WO 2018/009985 A1

(51) International Patent Classification:
G06F 17/50 (2006.01) *G05B 19/4097* (2006.01)
G06F 17/30 (2006.01) *E04G 21/22* (2006.01)

(21) International Application Number:
PCT/AU2017/050738

(22) International Filing Date:
17 July 2017 (17.07.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
2016902787 15 July 2016 (15.07.2016) AU

(71) Applicant: **FASTBRICK IP PTY LTD** [AU/AU]; 122 Sultana Road West, High Wycombe, Perth, Western Australia 6057 (AU).

(72) Inventor: **PIVAC, Mark**; 11 Keane Way, Lesmurdie, Perth, Western Australia 6076 (AU).

(74) Agent: **GOLJA HAINES & FRIEND**; Suite 7, 23 Richardson Street, South Perth, Perth, Western Australia 6151 (AU).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(54) **Title:** COMPUTER AIDED DESIGN FOR BRICK AND BLOCK CONSTRUCTIONS AND CONTROL SOFTWARE TO CONTROL A MACHINE TO CONSTRUCT A BUILDING

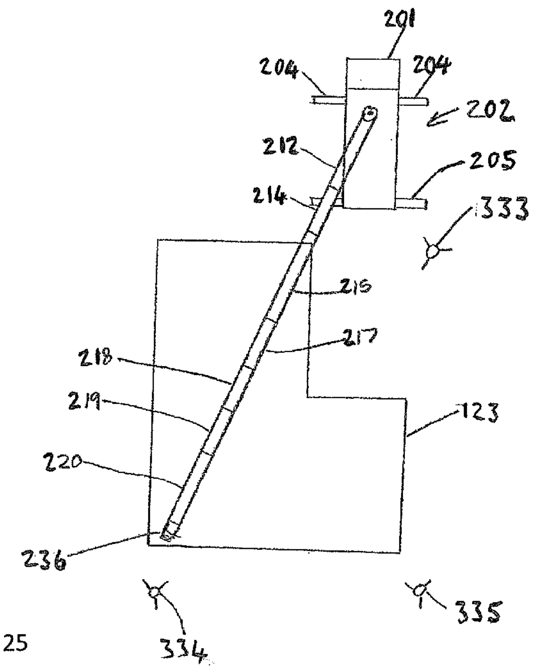

FIGURE 25

(57) **Abstract:** Computer aided design software for designing a building or other structure of brick construction, where in addition to the usual three dimensional modelling and rendering typical of CAD software, tabular data describing the spatial location and orientation of each brick is provided, including information regarding which bricks are cut to length so as to be shortened, and where they are located along each course, and which bricks are machined, drilled or routed for services or other special fittings. Data pertaining to this is compiled in a database for access by control software to control a brick laying machine to build a building or other structure from bricks. The database may receive via interface with a scanner data being a measure of the elevation of the footings and/or concrete pad that has been constructed according to the building plan and for each brick of the first course, to determine how much material must be machined off the bottom of each brick so that when the first course is laid, the tops of the bricks of the first course are at the same

[Continued on next page]

WO 2018/009985 A1

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

level. This machining data is stored for each brick with the tabular data produced by computer aided design software, so that the control software can control the brick laying machine to machine and cut each brick as per the stored data, and convey each brick to the stored position on the footing, pad or previously laid course of bricks, with application of adhesive prior to positioning of the brick.

Computer Aided Design for Brick and Block Constructions and Control Software to Control a Machine to Construct a Building

Technical Field

[0001] This invention relates to the construction industry, and in particular to the design of buildings of brick and block construction. This invention provides a computer software solution for designing buildings, and subsequently executing their construction from data compiled by the software.

[0002] Throughout the specification unless the context requires otherwise, the word "comprise" or variations such as "comprises" or "comprising", will be understood to imply the inclusion of a stated integer or group of integers but not the exclusion of any other integer or group of integers.

[0003] In this specification the word "brick" is intended to encompass any building element such as a brick or block, to be placed during the construction of a building or wall or the like. Block can encompass prefabricated beams that can form lintels to be incorporated into brick work.

Summary of Invention

[0004] In accordance with the invention there is provided computer aided design software for designing a brick construction, said computer aided design software having a user interface allowing a user to input building plan data, said computer aided design software generating from said building plan data, data representative of a scale top-plan view of walls with predetermined known wall length dimensions, said computer aided design software generating a virtual extrusion of length commensurate with the wall height;
said computer aided design software determining brick by brick placement data for said walls, including the identification of the position and orientation in three dimensions of each brick relative to a point of origin, including determining brick stepping and brick nesting at wall intersections, and having regard to a predetermined minimum cut brick length - determination of cut length data for individual bricks to be cut to length in order to meet the required dimensions of wall length;
said computer aided design software compiling a brick placement database including brick type, position and orientation in three dimensions of each brick relative to said point of origin, and cut length data for each brick identified as to be cut to length.

[0005] Preferably said computer aided design software has a first table containing one or more user selectable settings allowing stock brick type and size parameters to be stored and selected for walls. This may include stock brick type and size parameters for external walls and internal walls, where these differ.

[0006] Preferably said building plan data is representative of a scale top-plan view of external walls and any internal walls.

[0007] Preferably said computer aided design software generates footing data and/or concrete pad data including the dimensions, position and orientation of the footings and/or concrete pad relative to a point of origin, relative height off-set between the bottom of the external walls and optionally the bottom of the internal walls and between different sections of internal walls and optionally determines the required volume of concrete to form the pad.

[0008] Preferably said computer aided design software allows user input of and storing positioning data for voids and/or apertures in said extrusion, said voids and/or apertures being of dimensions commensurate with the height and width of doors and windows to be fitted in the completed building. The step of determination of cut length data for individual bricks to be cut to length in order to meet the required dimensions of wall length will include and wall segment length adjacent to voids and apertures. The cut length data for each brick identified as to be cut to length is stored in said brick placement database.

[0009] Preferably said computer aided design software allows user input of and storing services positioning data for at least one of plumbing, electrical and ICT (Information and Communication Technology) and sound and vision cabling and connection points in said external walls and in said internal walls; said computer aided design software generating positioning data for channels in said walls to carry services and recesses in said walls for said connection points, said computer aided design software generating machining data for the location of recesses and cavities to be machined in individual bricks according to the positioning data of said channels and recesses; and storing said machining data in said brick placement database.

[0010] It should be noted that the wall height may change around the building to allow for sloping roofs and ceilings and / or changes in floor level or ceiling height, so the wall height of internal walls and external walls need not be uniform.

[0011] Preferably said computer aided design software determines the order that each brick is to be laid, and creates in said brick placement database, an index number allocated to each brick, to identify the brick laying order. This order will start at the first course on the footings and/or pad, and then work upwards, course by course.

[0012] All user input may be via a graphical user interface.

[0013] The position and orientation data for each brick includes in its simplest form, x and y horizontal dimensions relative to said point of origin, with reference to a position in or on each brick, z vertical dimension, and Ø angle relative to magnetic north or other direction. The point of origin may be a position on the building site outside, in, or within the periphery of the pad. Other orientation data may include pitch angle and roll angle where the brick is to be laid in an orientation other than flat. The reference position on each brick is preferably the centroid or centre of the horizontal area of each brick, trimmed or whole, but for simplicity not taking into account routed cut-outs or recesses.

[0014] Most preferably the reference position on each brick is the centroid of all dimensions of each brick, trimmed or whole, but for simplicity not taking into account routed cut-outs or recesses. Where bricks are cut to length (shortened), clearly the centroid position will change.

[0015] The relative height off-set between the bottom of the external walls and the bottom of the internal walls may be reflected in different z values for the first course of bricks of the external course and the internal course. The difference in z values for the first course of bricks of the external course and the internal course arises from the type of damp course construction which determines the footings and/or concrete pad configuration.

[0016] The computer aided design software may take into account and allow for different spacings between adjacent bricks and thicknesses of adhesive or mortar between adjacent bricks. Where this occurs values for adjacent brick spacing A and brick base spacing B are stored in the brick placement database. The spacings may be global settings, or individual settings stored with each brick. In this manner, the required volumes of adhesive or mortar may be dispensed on the underside of the brick to provide the required brick spacing B and dispensed on the side or end of the brick to provide the required adjacent brick spacing A. Typical global settings for an A value can be up to 10 mm but typically might be 6 mm, and for a B value might be 0 mm. Where construction adhesive is used, the B value is zero, since the construction adhesive will be compressed as the brick is placed. Each brick except for the very first to be laid in a course will have accounted for, an A value and A-location data identifying where on the brick adhesive is to be applied to give effect to the A value.

[0017] With the method of the invention, each brick is allocated attributes pertaining to position in space (at least x, y, z and Ø values) and whether it is in original form or cut or otherwise machined, and if cut or machined, mathematical expressions defining the location of the volume of material to be excised from the brick with reference to said position on said brick.

[0018] If the brick is to be cut or routed, the design software exports a 3D model (typically a 3D DXF file, but could be and IGES or STL or other 3D file) of the brick which is then later used by a quality control system (QC system) which compares the 3D model to a 3D scan of the brick. The file name for the 3D model can be referenced in the brick placement database. In the control software to control a brick laying machine, a 3D scan of the machined brick can be made, either at the point where it is machined, or later immediately prior to it being laid, or both. A go/no go decision can be later made by the QC system in the control software, based on a percentage difference. This makes sure that the brick has been processed as required and hasn't had one or more pieces break off during machining (routing or cutting).

[0019] For cut or routed bricks the design software exports the coordinates of the cut or rout. Another routine of software takes those coordinates and processing type and creates G code data to program the router or saw for that particular cut. The G code data is associated in the brick placement table data for the brick, as a G code data field in the brick placement database.

[0020] Where there is a difference in mortar or adhesive thickness, each brick may also be allocated a B value, and most bricks have one A value and A-location data, with a very few bricks having two A values and A-location data for each A value.

[0021] Preferably the design software brick placement database includes at least one trim data field to store a trim value or a trim value array for each brick.

[0022] This trim data field may store values representative of the height of the surface for at least one location for each brick at which each brick is to be located according to said brick placement database. From this data the amount of material to be machined from the bottom of each brick can be determined. This trim data can be received from a scanner located at a surveying position, which measures the relative surface height of a surface extent where bricks are to be laid.

[0023] This data may be filled at the design stage, based on a separate scan of the surface where bricks are to be laid, and imported into the brick placement database for subsequent communication to the control software to control a brick laying machine to build a building or other structure of brick construction, according to the plan embodied in the brick placement database.

[0024] Alternatively, this data may be filled during the construction stage, based on a scan of the surface where bricks are to be laid by surveying equipment interfaced with the control software to control the brick laying machine, and the control software fills the trim data fields.

[0025] The trim data fields allow for the bedding surface that the brick will be placed on, to be scanned by surveying equipment, which may be operated independently or associated with an automated brick laying machine. The brick placement database may then include a directive variable that signals to the automated brick laying machine that the bedding area for the brick needs to be scanned and the brick may need to be trimmed to suit the bedding surface. The surveying equipment associated with the automated brick laying machine may then scan the bedding surface and determine the amount of material to be removed from the brick so that when the brick is laid, the top surface of the brick is at the correct Z height and is level. This may be, and is usually done for the first course of bricks whose bedding surface is the slab or footing which may be subject to manufacturing variation from the concrete pouring and finishing process. If it is known that the tolerance of the brick height (Z dimension) is large than it may be necessary to scan the bedding surface after a certain number of courses have been laid.

[0026] As an alternative to scanning the slab by the machine during or immediately prior to the brick laying process, optionally the bedding surface or slab data could be obtained from a previously acquired scan of the slab and imported to the design software. This is of particular use where a wall is to be built on a known prior surface which may have existing survey or scan data, such as an existing slab or footing or on civil works such as a wall on a bridge or a freeway road, such as a freeway sound wall. A trim surface, or trim value for each of the first course of bricks is then associated with each first course brick in a tabular file. The trim surface may optionally be defined as a trim value or a grid array of trim values or by a known CAD data exchange format such as STL or IGES, point cloud or as a toolpath file, for example a G code file which can be run by the automated brick laying machine (the router or the saw in the machine) to trim the brick.

[0027] As a consequence, an entire construction can be defined by brick type and attributes as a data set. The data set may be provided as an accessable database, or in in tabular form with setting out the brick data in the order that the bricks are laid. The data set may include by reference additional files such as trim surface files or G code files for machining operations.

[0028] Preferably said computer aided design software has a first library in which data pertaining to one or more building plans in the form of said data sets are stored.

[0029] Preferably said computer aided design software has a second library in which data pertaining to one or more pre-defined rooms are stored, said pre-defined rooms being selected from at least one kitchen, at least one bathroom, and optionally other rooms, each said at least one kitchen including allocated space for a sink, a cooking appliance, and a refrigerator, each said at least one bathroom including allocated space for at least one of a toilet, bidet, water closet, a bathtub, a shower and a hand basin or vanity, where using said GUI a said predefined room may be merged with said scale top-plan view.

[0030] Preferably said computer aided design software has a third library in which data pertaining to a plurality of doors and windows are stored, corresponding to data for stock items used in determining said positional data based on selected doors or windows.

[0031] Preferably said computer aided design software treats walls of a structure to be built as segments extending between intersections of brickwork, where each segment has a course segment extending between intersections of brickwork and window and door opening edges; where any said course segment has a length s, where:

$$s = n.(b + A) + r + A + p.e + p.A + q.f + q.A - A$$

where A is the A value or gap),

b is the stock brick length,

e and f are the end overlap at a brick wall intersection,

p may be 1 (indicating e is equal to the width of a brick at the intersecting wall, or zero (abutting),

q may be 1 (indicating f is equal to the width of a brick at the intersecting wall, or zero (abutting), r is the remainder which may be zero or greater than or equal to 0.2 b, preferably 0.25 b, and if this is true, a single brick is cut to length r to complete the course segment;

and if r is less than 0.2 b, preferably 0.25 b,

$$s = (n - 1).(b + A) + 2r + 2A + p.e + p.A + p.f + q.A - A$$

where r is the length that two bricks are cut to, to locate within and complete a course segment having n-1 bricks.

[0032] In order to achieve the preferred stepping, preferably where a said course segment includes two bricks of length r, the immediately overlying course segment includes a single brick of length r balanced on the join between the two bricks of length r, with two bricks cut to length of c=(b+r)/2, located on either side thereof, with bricks of stock brick length b continuing away from at least one of said two bricks cut to length of c. The course segment length can be described by the following:

$$s = (n - 2).(b + A) + r + A + 2(c+A) + p.e + p.A + p.f + q.A - A$$

[0033] Also in accordance with the invention, there is provided control software to control a brick laying machine to build a building or other structure of brick construction, said control software accessing a brick placement database compiled by computer aided design software as described above, said control software controlling said machine to cut and optionally to machine each said brick in accordance with data stored in said brick placement database, and controlling said machine to convey each said brick one by one, and apply adhesive and locate each said brick according to data stored in said brick placement database

[0034] Also in accordance with the invention there is provided control software to control a brick laying machine to build a building or other structure of brick construction, said control software accessing a brick placement database including brick type, position and orientation in three dimensions of each brick relative to a point of origin, cut length data for each brick identified as to be cut to length, and machining data for each brick including a trim value or a trim value array for each brick being trim data derived from data received from a scanner located at a surveying position to measure the relative surface height of a surface extent where bricks are to be laid; said control software correcting for the difference in height of the surface extent for each brick, from the lowest point and the highest point for each course of bricks and determining from said trim data the amount to be machined from a horizontal face of each brick so that the top of each course of bricks is level when laid; said control software controlling said machine to cut and machine each said brick in accordance with data stored in said brick placement database, said control software controlling said machine to convey each said brick one by one, and apply adhesive and locate each said brick according to data stored in said brick placement database.

[0035] Also in accordance with the invention there is provided control software to control a brick laying machine to build a building or other structure of brick construction, said control software accessing a brick placement database including brick type, position and orientation in three dimensions of each brick relative to a point of origin, cut length data for each brick identified as to be cut to length, and machining data for each brick; said control software including a scanner interface to receive data from a scanner located at a surveying position to measure the relative surface height of a surface extent where bricks are to be laid, storing the height of the surface for at least one location for each brick at which each brick is to be located according to said brick placement database; said control software correcting for the difference in positioning of said surveying position and said point of origin and determining the difference in height of the surface for said at least one location for each brick, from the lowest point and the highest point for each course of bricks and storing the difference in said brick placement database as trim data in the form of a trim value or trim value array for each said brick corresponding with the amount to be machined from a horizontal face of each brick so that the top of each course of bricks is level when laid; said control software controlling said machine to cut and machine each said brick in accordance with data stored in said brick placement database, said control software controlling said machine to convey each said brick one by one, and apply adhesive and locate each said brick according to data stored in said brick placement database.

[0036] The surface extent where bricks are to be laid may comprise footings and/or a concrete pad on which a building is to be constructed of bricks, or the upper surface of a structure upon which a course of bricks is to be laid.

[0037] Preferably, unless the computer aided design software that compiled the brick placement database has done so, said control software determines the order that each brick is to be laid, and creates an index number allocated to each brick, to identify the brick laying order, and enters the index number into the brick placement database.

[0038] Preferably said control software includes a library of handling identifiers which each identify a unique handling device within said brick laying machine, and said control software produces a handling table identifying individual bricks and associating individual bricks with a particular handling device at a particular time, and updating said handling table as individual brick progress from handling device to handling device with the elapsing of time.

[0039] Handling devices can include programmable brick handling apparatus to convey bricks from a pack of bricks, to a brick laying gripper mounted to a brick laying and adhesive applying head. The programmable brick handling apparatus may include dehacking robotic handlers that unpack rows of individual bricks from a pallet, and a series of devices each with grippers that handle individual bricks between the dehacking robotic handlers and the brick laying gripper, optionally via at least one brick machining tool, as pre-programmed. Where the bricks are moved out along a boom, there may be a plurality of shuttles, each with a gripper, and individual bricks are passed between shuttles. All of these transfers between the programmable brick handling apparatus are logged in the handling table, so that the handling table provides a record of which brick is where and when.

[0040] Consequently, if for any reason the brick laying machine must be paused for any reason, it may be restarted, and the correct brick will be laid in the correct position.

[0041] Further, if for any reason a brick is damaged during a machining (cutting or routing) operation, it may be discarded at the machining tool, and the handling table can be updated by reallocating brick identification numbers to the bricks preceding the damaged brick in the supply chain.

[0042] Where damage to an individual brick is not determined until it reaches a position closer to the brick laying gripper, where any said brick already in transit along said handling devices includes no machining data in said brick placement database, said handling table can be updated by reallocating brick identification numbers to the bricks preceding the damaged brick in the supply chain.

[0043] However, where any said brick already in transit along said handling devices includes machining data in said brick placement database, due to the individual bricks being laid in order, it becomes necessary for the control software to run the brick laying machine in reverse, restacking the bricks and storing their restacked position until there are no bricks having associated said machining data in transit along said handling devices, whereupon a replacement brick can be picked from said pallet and processed according to said brick placement database. Thereafter, any restacked bricks are picked up in order, and operation continues as pre-programmed.

[0044] Preferably said control software calculates corrected x y z position and orientation data relative to said point of origin for each brick recorded in said brick placement database, based on the difference between the location of the point of origin and the surveying position, and records the corrected x y z position and orientation data relative to said surveying position, for use in controlling said brick laying machine.

[0045] Thus controlled by the control software, the brick laying machine may build the building or structure, course by course, until completed to the required height. The first course of each course of bricks may be machined to reduce their height as necessary, according to the data from the scan measuring the relative surface height of footings and/or a concrete pad, so the tops of the first course of laid bricks are level. Thereafter, successive courses should not need their height adjusted by machining, in order to keep each course level. If the brick height tolerance is large then the top of a course may be scanned and the bottoms of the bricks in the subsequent course of bricks may be trimmed so that the top of the course of bricks is level.

[0046] Preferably said trim data is measured and stored as a trim value array for multiple locations for each said brick, so that said machine may machine said brick to compensate for localised footing or pad height excesses. Where greater attention is paid to the levelness of the footings or pad it will be sufficient to measure and store the trim value for four, three, two, or even one location for each said brick. In the ideal situation the slab or footing is accurate enough such that the trim data for all bricks is zero and no machining is necessary, however, in practice, unless greater expense has been incurred when laying the slab or footing, there will be variation in the level of the slab or footing.

[0047] The brick placement database contains position and orientation data for each brick, including in its simplest form, x and y horizontal dimensions relative to said point of origin, with reference to a position in or on each brick, z vertical dimension, and Ø angle relative to magnetic north or other direction. The point of origin may be a point on the building site outside of the periphery of the pad, or within the periphery of the pad. Other orientation data may include pitch angle and roll angle where the brick is to be laid in an orientation other than flat. The reference position on each brick is preferably the centroid or centre of the horizontal area of each brick, trimmed or whole, but for simplicity not taking into account routed cut-outs or recesses. Most preferably the reference position on each brick is preferably the centroid of all dimensions of each brick, trimmed or whole, but for simplicity not taking into account routed cut-outs or recesses. Where bricks are cut to length (shortened) the centroid position will change.

[0048] The relative height off-set between the bottom of the external walls and the bottom of the internal walls may be reflected in different z values for the first course of bricks of the external course and the internal course. The difference in z values for the first course of bricks of the external course and the internal course arises from the type of damp course construction which determines the concrete pad configuration.

[0049] The trim data may be represented as an adjusted z value for each brick and may include multiple z values tied to separate x and y values for each brick, to provide complex machining data where the machining is to adjust for localised undulations in the footing or pad.

[0050] The control software may take into account and allow for different spacings between adjacent bricks and thicknesses of adhesive or mortar between adjacent bricks. Where this occurs, values for adjacent brick spacing A and brick base spacing B are stored with each brick in the brick placement database. The control software interfaces with the adhesive or mortar applicator in the machine, to control the applicator to dispense the required volumes of adhesive or mortar on the underside of the brick to provide the required brick spacing B and on the side or end of the brick to provide the required adjacent brick spacing A. Each brick except for the very first to be laid in a course will have an A value and A-location data identifying where on the brick adhesive is to be applied to give effect to the A value.

[0051] With the methodology of the invention, each brick is allocated attributes pertaining to position in space (at least x, y, z and Ø values) and whether it is in original form or cut or otherwise machined, and if cut or machined, mathematical expressions defining the location of the volume of material to be excised from the brick with reference to said position in or on said brick. Where there is a difference in mortar or adhesive thickness, each brick may also be allocated a B value, and most bricks have one A value and A-location data, with a very few bricks having two A values and A-location data for each A value.

[0052] As a consequence an entire construction can be defined by brick type and attributes as a data set. The data set may be provided in the form of an accessable database or in tabular form, setting out the brick data in the order that the bricks are laid.

[0053] It should be noted that where this description refers to a database, this function may be performed by one or more data tables and such tables may be formed by a plurality of tables that cross reference each other.

[0054] Also in accordance with the invention there is provided a method of building a structure from bricks, comprising steps of determining the size of brick to be utilised; creating a scale top-plan view of walls with predetermined known wall length dimensions; determining brick by brick placement data for said walls, including identification of the position and orientation in three dimensions of each brick identification of the position in three dimensions and orientation of individual bricks to be cut to length in order to meet the required dimensions of wall length, and determining the order that each brick is to be laid, and storing this data in a brick placement database; measuring the relative surface height of a surface extent where bricks are to be laid , recording the height of footings for at least one location at which each brick is to be located according to said brick placement data, determining the difference in height of the footings for at least said one location for each brick, from the lowest point to the highest point for at least the first course of bricks and storing the difference from the lowest point as trim data in the form of a trim value or trim value array for said at least one location for each said each brick corresponding with the amount to be machined from a horizontal face of each brick so that the top of at least the first course is leveled when laid, the trim data being stored with said brick placement data; cutting each said brick in accordance with said brick placement data, conveying each said brick one by one, and applying adhesive and locating each said brick according to said brick placement data.

[0055] Also in accordance with the invention there is provided a method of building a building or other structure, comprising steps of determining the size of brick to be utilised for external walls and the size of brick to be utilised for internal walls; creating a scale top-plan view of external walls and internal walls with predetermined known wall length dimensions, determining footing and/or concrete pad data including the dimensions of the footings and/or concrete pad, relative height off-set between the bottom of the external walls and the bottom of the internal walls and between different sections of internal walls and optionally determining the required volume of concrete to form the pad; planning the configuration of the walls from the footings and/or pad up to the tops of the walls including positional determination of voids of dimensions commensurate with the height and width of doors and windows to be fitted, and positioning data for channels and recesses (chasing) for services of plumbing, electrical and ICT and sound and vision cabling and connection points in said external walls and in said internal walls; determining brick by brick placement data for said external walls and said internal walls, including identification of the position and orientation in three dimensions of each brick relative to the footings and/or concrete pad, identification of the position in three dimensions of individual bricks to be cut to length in order to meet the required dimensions of wall length, void size and aperture size, and generating machining data for the position of recesses and cavities to be machined in individual bricks according to the positioning data of said channels and recesses, and determining the order that each brick is to be laid, and storing this data in a brick placement database; measuring the relative surface height of a surface extent where bricks are to be laid , recording the height of the footings and/or a concrete pad for at least one location at which each brick is to be located according to brick placement data as described above, determining the difference in height of the pad for at least said one location for each brick, from the lowest point to the highest point for at least the first course of bricks and storing the difference from the lowest point as trim data in the form of a trim value or trim value array for said at least one location for each said each brick corresponding with the amount to be machined from a horizontal face of each brick so that the top of at least the first course is level when laid, the trim data being stored with said brick placement data; cutting and machining each said brick in accordance with said brick placement data, conveying each said brick one by one, and applying adhesive and locating each said brick according to said brick placement data.

[0056] The surface extent where bricks are to be laid may be footings and/or a concrete pad on which a building is to be constructed of bricks or the upper surface of a structure on which bricks are to be laid.

[0057] The building or structure may be built course by course until completed to the required height. The first course of each course of bricks is machined to reduce their height as necessary, according to the data from a scan measuring the relative surface height of footings and/or a concrete pad so the tops of the first course are level. Thereafter, successive courses should not need their height adjusted by machining, in order to keep each course level, unless brick size tolerance is so poor that a rescan of the tops of completed courses is required during construction.

[0058] Preferably said trim data is measured and stored as a trim value array for multiple locations for each said brick, so that each said brick is machined to compensate for localised footing or pad height excesses. Where greater attention is paid to the levelness of the footings or pad it will be sufficient to measure and store the trim value for four, three, two, or even one location for each said brick.

[0059] The position and orientation data for each brick includes in its simplest form, x and y horizontal dimensions relative to a reference point, with reference to a position on each brick, z vertical dimension, and Ø angle relative to magnetic north or other direction. The reference point may be a point on the building site outside of the periphery of the pad, or within the periphery of the pad. Other orientation data may include pitch angle and roll angle where the brick is to be laid in an orientation other than flat. The reference position on each brick is most preferably the centroid or centre of the horizontal area of each brick, trimmed or whole, but for simplicity not taking into account routed cut-outs or recesses.

[0060] The relative height off-set between the bottom of the external walls and the bottom of the internal walls may be reflected in different z values for the first course of bricks of the external course and the internal course. The difference in z values for the first course of bricks of the external course and the internal course arises from the type of damp course construction which determines the concrete pad configuration.

[0061] The method may take into account and allow for different spacings between adjacent bricks and thicknesses of adhesive or mortar between adjacent bricks. Where this occurs values for adjacent brick spacing A and brick base spacing B are stored with each brick, so the required volumes of adhesive or mortar may be dispensed on the underside of the brick to provide the required brick spacing B and dispensed on the side or end of the brick to provide the required adjacent brick spacing A. Each brick except for the very first to be laid in a course will have an A value and A-location data identifying where on the brick adhesive is to be applied to give effect to the A value.

[0062] With the method of the invention, each brick is allocated attributes pertaining to position in space (at least x, y, z and Ø values) and whether it is in original form or cut or otherwise machined, and if cut or machined, mathematical expressions defining the location of the volume of material to be excised from the brick with reference to said position on said brick. Where there is a difference in mortar or adhesive thickness, each brick may also be allocated a B value, and most bricks have one A value and A-location data, with a very few bricks having two A values and A-location data for each A value.

[0063] As a consequence an entire construction can be defined by brick type and attributes as a data set. The data set may be provided in tabular form with setting out the brick data in the order that the bricks are laid. During construction the controller adds data to the data set (database). This added data includes the position of the brick in the machine and photo and scan data and offset data. For example as a brick moves from a stick clamp to another, the database is updated with the coded location of the brick in its updated current clamp. When the brick is QC scanned, the scan data is stored in a unique file and the file name is associated to the database. At various locations on the machine, computer vision photographs are taken of the brick to determine its position so that an offset can be applied to the clamp that will next handle the brick, so that the brick is grasped in the correct position. Each of these photographs is stored in a file and the file name is associated to the database. This is done so that if there is a problem, such as a brick being laid incorrectly, the history of the brick as it passed through the machine can be reviewed for the purpose of fault finding for machine maintenance, or to give feedback to brick manufacturers or transporters if the bricks are not of acceptable quality or have damage or defects. The stored data also becomes part of the Quality Assurance records for the build.

[0064] In the most preferred embodiment, the invention comprises three aspects, first a computer aided design software for designing a building or other structure of brick construction, where in addition to the usual three dimensional modelling and rendering typical of CAD software, tabular data describing the spatial location and orientation of each brick is provided, including information regarding which bricks are cut to length so as to be shortened, and where they are located along each course, and which bricks are machined, drilled or routed for services or other special fittings. This data is exported to a database which may be accessed by control software to control a brick laying machine.

[0065] The second aspect comprises control software to control a brick laying machine to build a building or other structure from bricks. The database may receive data from a scanner which measures the elevation of the footings and/or concrete pad that has been constructed according to the building plan and for each brick of the first course, to determine how much material must be machined off the bottom of each brick so that when the first course is laid, the tops of the bricks of the first course are at the same level. This machining data is stored for each brick with the tabular data produced by computer aided design software, so that the control software can control the brick laying machine to machine and cut each brick as per the stored data, and convey each brick to the stored position on the footing, pad or previously laid course of bricks, with application of adhesive prior to positioning of the brick.

[0066] The third aspect of the invention comprises the combined method of building a building or other structure of brick construction, including steps of design of the building, determination of location of bricks including milling and cutting to size of individual bricks, scanning the footings and/or pad for height variations, adjusting the milling data, and then carrying out predetermined milling and cutting operations on each brick as necessary, prior to placing each brick.

Brief Description of Drawings

[0067] Preferred embodiments of the invention will now be described with reference to the drawings, in which:

Figure 1 is an orthographic projection of the external perimeter wall of a house;

Figure 2 is an orthographic projection of the internal perimeter wall of a house;

Figure 3 is a top plan view of the external perimeter wall of a house;

Figure 4 is tabular data showing stock brick parameters;

Figure 5 is an orthographic projection of part of a first external course of bricks;

Figure 6 is an orthographic projection of part of the first external course of bricks of figure 5 and part of the second external course of bricks;

Figures 7 to 10 are orthographic projections of external brick nesting configurations;

Figure 11 is a top plan view of the external perimeter wall of the house shown in figure 3 with the internal walls added;

Figure 12 is a broken isometric view of part of the first course of an internal wall, running in a north/south or longitudinal direction;

Figure 13 is a broken isometric view of part of the first course of an internal wall, running in an east/west or transverse direction;

Figure 14 is an isometric view showing internal wall brick stepping for a bay length of up to twice the brick length with associated vertical gaps;

Figure 15 is an isometric view showing internal wall brick stepping for a bay length of more than twice the brick length with associated vertical gaps;

Figure 16 is an isometric view showing one possible corner nesting configuration;

Figure 17 is an isometric view showing one possible tee junction nesting configuration;

Figure 18 is an isometric view showing one possible cross nesting configuration;

Figure 19 is an isometric view showing one possible end stepping configuration;

Figure 20 is an isometric view from below showing one possible tee junction nesting configuration with a rebate routed into a lower course external brick;

Figure 21 is a vertical cross section through part of a wall and slab of a building showing the wall and footing configuration;

Figure 22 is a vertical cross section through part of a wall and slab of a building showing an alternative wall and footing configuration to that shown in figure 21;

Figure 23 shows a view of the automated brick laying machine 202 with its truck base 201 with the boom and stick assembly 341 unfolded;

Figure 24 shows a view of the automated brick laying machine 202 with the boom and stick assembly 341 folded and stowed for driving on a public road;

Figure 25 shows a site plan of the automated brick laying machine 202 set up near a concrete slab 123 on which the automated brick laying machine 2 will build a structure;

Figure 26 shows a plan view of the automated brick laying machine 202;

Figure 27 shows a side view of the boom assembly.

Description of Embodiments

[0068] The computer aided design software according to the embodiment can be implemented as a fully featured CAD program, or as a plug-in for an existing CAD program, such as Solidworks. The control software to control a brick laying machine to build a building which is designed in the computer aided design software imports a brick placement database produced in the computer aided design software, and then the brick laying machine implements the construction of the building according to the data contained in the brick placement database. During the construction phase, the brick placement database may have fields (cloumns) added to it, in order to associate new data with each brick.

[0069] The computer aided design software allows the user, once size of brick to be utilised for external walls and the size of brick to be utilised for internal walls has been determined, to create a scale top-plan view of external walls and internal walls with known wall length dimensions as determined by the user in accordance with the requirements of the person commissioning the build. The user determines the footing and/or concrete pad data including the dimensions of the footings and/or concrete pad, and relative height off-set between the bottom of the external walls and the bottom of the internal walls and between different sections of internal walls.

[0070] The computer aided design software can, through inputting of concrete thickness required to meet load bearing capacity of the built structure, determine the required volume of concrete to form the pad.

[0071] The computer aided design software provides for the user to plan the configuration of the walls from the footings and/or pad up to the tops of the walls including positional determination of voids of dimensions commensurate with the height and width of doors and windows to be fitted, and positioning data for channels and recesses (chasing) for services of plumbing, electrical and ICT and sound and vision cabling and connection points in said external walls and in said internal walls.

[0072] The computer aided design software then determines brick by brick placement data for the external walls and the internal walls, and identifies the position and orientation in three dimensions of each brick relative to the footings and/or concrete pad, and identifies the position in three dimensions of individual bricks to be cut to length in order to meet the required dimensions of wall length, void size and aperture size.

[0073] The computer aided design software generates machining data for the position of recesses and cavities to be machined in individual bricks according to the positioning data of said channels and recesses.

[0074] The computer aided design software then determines the order that each brick is to be laid, based on a rule that requires any brick extending across an intersection or to the apex of a corner to be laid first. A blockdependency data field can be provided, associated with each brick in order to identify a set of bricks that must be laid before the brick specified. This is to avoid laying arm collisions with previously laid bricks.

[0075] A brick placement database is compiled, containing data identifying the position and orientation in three dimensions of each brick relative to a datum point which relates to a point on the footings and/or concrete pad, and where bricks are trimmed to length, the trimmed length, and the order in which the bricks are to be laid. All position and orientation data is referenced to a central point in each brick, whether trimmed to length or whole. The gap to the next brick is also stored as an A value in the brick placement database. Machining data for services, or any special arrangements of interconnecting walls is also stored in the brick placement database, against each brick. Blockdependency data is included for each brick, where necessary.

[0076] For each side of each brick requiring routing or cutting, a record is placed in the brick placement database. Each record provides a link to the brick it is associated with, cross-referencing to additional tables that contain the actual routing and cutting information in a form that can be read by the routing tool or saw, as the case may be.

[0077] Additional fields in the brick placement database are also provided for, against each brick, which can be populated at the design stage when the footings and/or slab have been poured and scanned for surface height variation, or can be populated during the construction phase, by input to the control software for controlling the brick laying machine. These additional fields include trim data containing a trim value array for each brick, which is the amount of material that must be machined from the bottom of each brick, so that when the bricks are all laid, the tops of the course will be level.

[0078] A laser scanning device is arranged to measure the relative surface height of a surface extent where bricks are to be laid, recording the height of the footings and/or a concrete pad, or the height of an existing course of bricks or other surface, at the location where each brick is to be located according to brick placement database as described above, to determine the difference in height of the surface across the location for each brick, from the lowest point to the highest point for the first course of bricks and stores the difference from the lowest point as trim data in the form of a trim value array for each brick. This data corresponds with the amount to be machined from a horizontal face of each brick so that the top of each course is level when laid, the trim data being stored with said brick placement data.

[0079] Special considerations regarding the execution of a building plan in the computer aided design software, including location of trimmed bricks and nesting of bricks will now be discussed.

[0080] As will be apparent from the discussion that follows, each course of each wall of a building or structure has a start point and a finish point. The course distance between start point and end point for each course is known, determined from the scale top-plan view produced in the computer aided design software. These points comprise two of the end of a course of bricks, the stepped end of a course of bricks plus adhesive thickness, the edge of a window and the edge of a door. A stepped end of a course of bricks is stepped inwards along the length of the course by the wall thickness of an intersecting brick of an intersecting course, usually forming part of a wall running at 90 degrees to the course. The minimum brick length is determined by the length of the gripper of the brick laying machine, which in the present embodiment is 120 mm. It would be impractical for the brick laying machine to handle bricks of shorter length than its gripper as this would invariably lead to the gripper clashing with bricks that had already been placed.

[0081] A course is made up of a number of stock bricks of known length, including where necessary, one or more bricks that are trimmed in length, referred to as a remainder which can be no shorter than 120 mm. Where two or three trimmed bricks are utilised and the lengths of the trimmed bricks is determined to ensure that brick stepping conforms with accepted practices. In the embodiment, the external bricks are 258 mm high, 500 mm long and 240 mm deep. The internal bricks are 258 mm high, 500 mm long and 115 mm deep.

[0082] The first course is made up of a number of stock bricks of length 500mm divided into the length of the wall. The remainder is determined from a distance greater than 125 and less than 500.

[0083] These bricks are illustrated in the figures 7 to 10, and 12 to 24 and are provided with a tongue and groove configuration in their ends to assist as light blockers to prevent light from being seen on the opposite side.

[0084] The computer aided design software allows the CAD operator to sketch the plan of the building, in this example, a house. Referring to figure 1, the CAD operator draws the external perimeter 11 of the exterior wall 13 of the house as straight line sketch segments 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, and closes the sketch pattern to make it a closed contour. The external perimeter 11 is drawn using a 125 mm grid system to position the walls at 125 mm increments. With this arrangement the bricks can be cut into 125, 250, 375 to fill the remainder gap to ensure that brick stepping conforms with accepted practices.

[0085] External openings for doors and windows are closed. Where the entire structure is built on a concrete pad, the external perimeter will represent the concrete pad dimensions. For a level concrete pad the elevation of the pad is given a z value of 0. The lowermost elevation of the wall should be set to the same z value. Alternatively for a stepped down damp course building construction, the external wall may be set down. The parallel lines of the wall in figure 1 represent the exterior 59 of the external wall and the interior 61 of the external wall. Each sketch segment 15 to 57 represents a wall from end to end.

[0086] The next step is to draw the internal walls 63 as a single line which traces the centrelines of the bricks whilst maintaining 125mm grid system to position the walls at

125mm increments. Referring to figure 2, the dashed lines 65 either side of the single line, denote the surface of the interior bricks. The lowermost elevation of the internal walls should be set to a y value of 0.

[0087] Referring to figure 3, the exterior wall each sketch segment 15 to 57 on the perimeter outer contour are collected and ordered in an anti-clockwise direction starting from the closest to the projects origin 63. Each sketch segment has a 3D x y z start point and 3D x y z finish point. Segment 15 has its 3D x y z start point at origin 63 and its 3D x y z finish point at the junction 65 with segment 17.

[0088] The sketch segments may be drawn in any direction, but the software collects the sketch segments and swaps the start and finish point (if required as it may be pointing in the opposite direction) on the sketch segment to maintain the chain pattern from start to finish on each sketch segment in an anti-clockwise direction, with sketch segment 57 having its 3D x y z finish point back at origin 63.

[0089] Wall assemblies are added to the project and are attached to the start point of the sketch segment, then the brick component from the library containing the brick data is inserted into the brick wall assembly. Brick data includes the stock external wall brick depth (thickness), length and height, the external vertical brick gap, the external wall horizontal brick gap, the stock internal wall brick depth (thickness), length and height, the internal vertical brick gap, and the internal wall horizontal brick gap. Figure 4 shows representative data for these values.

[0090] All walls are created from a pattern of the first and second courses to the cap.

[0091] Referring to figure 5, the first (bottom) course 66 of bricks is shown. To maintain consistency, the first brick 67 of each wall is offset 69 on the bottom course the depth of the first brick and then is placed from this point based on the brick depth plus vertical gap (indicated at 69) from the brick data in figure 4. This offset 69 provides space to accommodate a brick from the intersecting course, plus the vertical gap which is to be filled with adhesive. The first brick 67 and last brick 71 in the sequence for the bottom course are full uncut bricks. Proceeding from the first brick 67, full uncut bricks 73 are allocated with the vertical gap, until a space referred to as a remainder gap 75 which exceeds the length of a full uncut brick, but is less than the length of two full uncut bricks plus spacing for three vertical gaps, is reached along the course 66. Two stepping adjustment bricks 77 of equal length, equal to the remainder gap less three vertical gaps, all divided by two, are allocated to the remainder gap 75. There is a right end adjustment 79 at the end of last brick 71 which is set to zero if the end of last brick 71 is a doorway or external corner, and set to 6 mm (the same as the vertical spacing) if the last brick 71 forms an internal corner.

[0092] For external walls, from corner to corner, or from door to corner or visa-versa, the wall length is equally divisible by 125.

[0093] The first (bottom) course starts flush on the end of the first corner and patterns as full bricks to the remainder gap minus the end of the perpendicular bricks depth, this being Right End Adjustment. The remainder is stepped using cut bricks of 125mm intervals. ie 125, 250, 375.

[0094] Doors are treated similarly as a corner nest where the perpendicular brick in the Right End Adjustment is cut to a 250 brick on either side of the door for each course respectively.

[0095] The bricks 67, 73, 75 and 71 are placed on the exterior 59 line being the outer edge of the external brick and lines up with the sketch segments.

[0096] Referring to figure 6, the second course 81 brick placement starts with a full brick 83 on the left end with no left end adjustment and the right end with a brick depth + vertical gap adjustment 85. There are two full bricks 87 to the right of the left end full brick 83, and three cut to length stepping adjustment bricks 89, 91 and 93. The second stepping adjustment brick 91 is the same length as one of the first course 66 stepping adjustment bricks 77. The first stepping adjustment brick 89 length is determined by half the length of a first course 66 stepping adjustment brick 77, added to the distance between the right hand end of brick 73 and the right hand end of brick 87. The third stepping adjustment brick 93 length is measured from the end of the second stepping adjustment brick 91 minus the brick depth + vertical gap adjustment 85.

[0097] Any course of bricks running between an intersection of bricks, a corner, or an edge for a window or a door, can be considered to be a course segment. Each course segment is populated in the design software with a number of full length bricks 73 extending from end adjustment 69 to end adjustment 79 (if any), with at least one full length brick 67, 71 abutting each end adjustment 69, 79 respectively, leaving a remainder gap 75. Where the remainder gap 75 is less than the minimum allowable remainder of 120 mm (or 125 mm under the 125 mm grid system), the calculated remainder gap 75 added to the length of a stock brick added to the A value for each, all divided by two, determines the length of two bricks 77 to fill the remainder gap 75, as is illustrated in figure 5 . End adjustments may include cut bricks to ensure proper nesting, particularly where the end of a course segment is an opening for a door or window.

[0098] Where the remainder is the same as or greater than the allowable remainder, the trimmed brick length to fill the remainder is the determined remainder size. All data pertaining to this is stored in the brick placement database.

[0099] Referring to figure 6, in the case where two remainder bricks 77 are utilised in a course, the immediately overlying course 81 includes a trimmed brick 91 of the same length as one of the bricks 77, balanced across the join between the two bricks 77 (that is to say with half of the brick 91 lying to either side of the join between the bricks 77). Two bricks 89 and 93 are cut to a length equal to the length of brick 91 plus a stock brick 87, all divided by two, and these trimmed bricks 89 and 93 are placed either side of brick 91. Standard stock bricks 87 then extend away from these trimmed bricks.

[00100] The second course starts with the Left End Adjustment, being the perpendicular bricks depth, then patterns as full bricks until a remainder gap minus a full brick on the end sitting flush with the external face of the wall occurs. The remainder must be greater than or equal to 125 mm and can consist of 2x375 mm, 2x125 mm, 1x125 mm cut bricks and consideration of the first courses remainder for stepping as described above ensures a vertical gap overlap occurs to ensure that there are no continuous vertical gaps occurring between courses.

[00101] The final brick on the second course is a full brick and sits flush with the external face of the returning wall.

[00102] The third course placed on top of the second course 81 is a repeat of the first course 66, and the fourth course placed on top of the third course is a repeat of the second course 81.

[00103] The order of laying bricks in each of the first and second courses (and in consequence the courses that follow) may be swapped so that efficiencies can be gained from nesting arrangements at door and window headers with lintels.

[00104] Referring to figures 7 to 10, various nesting configurations of bricks for external brick junctions are shown. Figure 7 shows the nesting configuration for a typical external corner in external bricks. The groove ends 101 and tongue ends 103 of the bricks are shown. Figure 8 shows the nesting configuration for a typical internal corner in external bricks. Figure 9 shows a nesting configuration for an external tongue end, and figure 10

shows a nesting configuration for an external groove end, both featuring cut bricks 105 at the ends. These are typically used for doors and windows.

[00105] Referring to figures 16 to 20, various nesting configurations of bricks for external brick junctions are shown. Figure 16 shows the nesting configuration for a typical internal corner in internal bricks. The groove ends and tongue ends of the bricks are shown. Figure 17 shows a nesting configuration for an internal corner in internal bricks. Figure 18 shows a possible nesting configuration for an internal wall intersection. Figure 19 shows a nesting configuration for an internal groove end, typically used for doors and windows. Figure 20 shows internal brick junction nesting of two external walls and their junction with an internal wall. Machining of the intersecting external brick to accommodate part of the lower course internal brick can be seen.

[00106] Another design consideration for the design software is the order of the laying of the bricks, which is another parameter included in the brick placement database. As any course segment is laid, the first brick to be laid is one that extends across a brick junction or to the apex of a brick junction corner. This is so that the gripper of the brick laying machine has access to lay the bricks. If a brick abutting such a brick was to be laid first, the gripper of the brick laying machine would not have access to be able to lay the brick that extends across a brick junction or to the apex of a brick junction corner.

[00107] Internal wall creation is similar to external wall creation, except that the brick is placed centrally to the sketch segments and not on a perimeter sketch segments. The stepping of the bricks is also different in that the end conditions (nesting of the bricks from 1 wall to another) have many possibilities. A rule of thumb has been applied based on the direction of the walls to which the left and right end conditions are applied. This rule of thumb is applied to allow for corners to nest in any situation.

[00108] In a wall heading in the direction from North to South or South to North as shown in figure 12 (considering the plan in Figure 11 and not magnetic or true north at the actual building site) the first course first and last bricks are extended past the corner brick and the second course brick is positioned internally, in order to lie flush against the intersecting brick. In a wall heading in the direction West to East or East to West, as shown in figure 13, the first and last bricks are opposite to North to South or South to North to allow for nesting of internal corners of the two intersecting courses.

[00109] Referring to figure 2, each entire line between distal points is considered to be a sketch segment. The line between points 107 and 109 is one such sketch segment. The internal wall is divided at sketch points 111, where intersected by other internal walls,

into separate bays 113. The brick stepping for each bay 113 is based on the distance and the number of full bricks that can be patterned into the length of the bay, the adjustment cut bricks and left and right adjustments for each bay based on nesting of other walls. For the determination of brick stepping, the bay length excluding any left and right end adjustment offset lengths is considered. Left and right end adjustment offsets are the thickness of bricks of intersecting walls at corner nesting configurations, as discussed for the external walls in the description above, but with dimensions adjusted for the different 125 mm depth of the internal bricks. Brick stepping is determined depending on whether the bay length (excluding the left and right end adjustment dimensions) is from the minimum brick length to the length of two bricks plus the vertical brick gap or whether the bay length exceeds the length of two bricks plus the vertical brick gap. See figure 14 for typical brick stepping where the bay length is up to the length of two full bricks plus the vertical gap, and figure 15 for where the bay length exceeds the length of two bricks plus the vertical brick gap.

[00110] The adjusted length of the stepping adjustment bricks 77, the centre xyz coordinates for each brick in each course of bricks, and the orientation of each brick, relative to the origin 63 together with the left end adjustment 69 and right end adjustment 79 are stored in a brick placement database, which defines the parameters for the bricks to be laid. Since the bricks have a tongue at one end and a groove at the opposite end, and they are laid in a straight line with tongue projecting into groove, the orientation of each brick runs at any value from 0 degrees to 359 degrees, to retain the tongue-groove co-operation. In addition, the 3D x y z start point and 3D x y z finish point for each sketch segment is stored, and the course number for each brick is stored, for example 0 for the first course, 1 for the second course, and so on. The order of laying the bricks, typically from the origin 63, is also stored. The first brick to be laid will have adhesive applied to its underside, and the bricks that follow will have adhesive applied to both their underside and the end (or in the case of a corner, a part of the side that abuts the previously laid brick), and the location of applied adhesive is stored in the brick placement database.

[00111] Each bricks 3D location point, length, cut, routing, chasing and rotation data is exported from the design software in the brick placement database, to the control software of the brick laying machine. The three dimensional coordinates of each brick are the length x, the depth or width y and the height z. These values are imported from the brick data contained in a first table containing stock brick sizes, or where the brick is trimmed, are calculated from a virtual bounding box for the brick, generated by the

design software. The orientation of the brick is measured against the design software project front plane running in the South to North direction and is degrees from 0 to 359

[00112] Where a brick is to be shortened it can be cut to leave the tongue end to be used, or the groove end. This will determine the handling of the brick for the cutting operation. The end to be used is predetermined and its data is stored in the brick placement database. The off cut length is calculated by the design software and the off cut is recorded as available stock for use elsewhere in the plan, and control software is programmed to retrieve an offcut brick portion from recorded available stock, with data pertaining to this being recorded in the brick placement database.

[00113] Figures 21 to 24 show various configurations of wall construction including the footing arrangements and finishing courses. Figure 21 shows a cross section through a wall 121 and slab 123 showing the bottom course 125 with a damp course building method and the top course 127 using an internal brick.

[00114] Referring to the section shown on figure 22, this shows the bottom course 125 sitting flush on the concrete slab 123 as alternative design, whereas referring to the section shown on figure 23, the bottom course 125 is one course lower than the surface 129 of the slab 123 forming a damp course ledge 131 around the perimeter of the concrete slab, used for external walls only.

[00115] Referring to the section shown on figure 24, the top course 127 on the external wall 121 is an internal brick 127 sitting flush on the internal face 133 of the wall.

[00116] The control software for controlling the brick laying machine is incorporated into control electronics in a control cabinet 282, to control the operation of a brick laying machine 202.

[00117] Referring to figure 24, a truck 201 supports a brick laying machine 202 which is mounted on a frame 203 on the chassis (not shown) of the truck 201. The frame 203 provides additional support for the componentry of the brick laying machine 202 beyond the support that would be provided by a typical truck chassis. The frame 203 has horizontal outwardly extending legs 204 and 205, each with hydraulically actuated push down legs 206 to stabilise the bricklaying machine, once it is parked in position for work.

[00118] Referring also to figure 26, the frame 203 supports packs or pallets of bricks 252, 253, which have been loaded onto a fold down rear deck 208 and moved onto dehacker bays 249 and 250 by scrapers 257. De-hacker robots (not shown) can take rows of bricks off the pallets and place them on a platform 251. A transfer robot (not

shown) can then pick up an individual brick and move it to, or between either a saw 246 or a router 247 or a carousel 248. The carousel 248 is located coaxially on a slewing ring 211 with a tower 210, at the base of the tower 210. The carousel 248 rotates to align to the tower to transfer the brick to a gripper on a shuttle running up the tower 210 to allow transfer of the brick to an articulated (folding about horizontal axis 216) telescoping boom.

[00119] The telescoping boom comprises telescopic boom elements 212, 214 and telescopic stick elements 215, 217, 218, 219, 220. Each element 212, 214, 215, 217, 218, 219, 220 of the folding telescoping boom has a shuttle located inside on a longitudinally extending track in the element, to transport a brick along the longitudinal extent of the element. The bricks are moved through the inside of the folding telescoping boom by the linearly moving shuttles. The shuttles are equipped with grippers that pass the brick from shuttle to shuttle. The shuttles in the telescoping elements are located alternately along the top of one element and along the bottom of the next element, when viewed with the boom unfurled, as shown in figures 25 and 27. The shuttles have grippers to grip a brick as discussed above, and since the shuttles run on opposite sides of immediate next telescoping elements, when the shuttles are coincident both sets of jaws on both shuttles can grip the brick, allowing the brick to be passed from one shuttle to the other.

[00120] The shuttles in the elements 214 and 215 run along the top of these elements and a pivoting gripper is provided about axis 216, so that a brick can transfer from the gripper on the shuttle in element 214 to the gripper on the axis 216 which can rotate to align to the gripper on the shuttle in element 215.

[00121] A pivoting gripper is also provided on the axis 213 about which boom element 212 mounts to the tower 210, so that a brick can transfer from the gripper on the shuttle on the tower 210, to the pivoting gripper on the axis 213 and then to the gripper on the shuttle running along the bottom of element 212.

[00122] The carousel 248 also has a pivoting gripper 274 into which a brick is placed by the transfer robot, before the carousel 248 rotates and the pivoting gripper 274 thereon pivots to present the brick to the grippers on the shuttle on the tower 210.

[00123] The end of the boom is fitted with a brick laying and adhesive applying head 232. The brick laying and adhesive applying head 232 mounts by pins (not shown) to element 220 of the stick, about an axis 233 which is disposed horizontally. The poise of the brick laying and adhesive applying head 232 about the axis 233 is adjusted by double

acting hydraulic ram 235, and is set in use so that the tracker component 330 is disposed uppermost on the brick laying and adhesive applying head 232. A gripper is mounted about the pivot axis 233 and received a brick from the shuttle at the end of stick element 220, flips it and presents it to the brick laying and adhesive applying head 232, which applies adhesive to the brick and presents it to a robot 236 with a gripper 244 that lays the brick. Vision and laser scanning and tracking systems 334, 335, 333 are provided to allow the measurement of as-built slabs 123, bricks, the monitoring and adjustment of the process and the monitoring of safety zones. The first, or any course of bricks can have the bricks pre-machined by the router module 247 so that the top of the course is level once laid, as is discussed above.

[00124] The transfer robot, the saw 246, and the router 247 each have a gripper that can hold a brick at any point in time, as do the grippers on the carousel 248, the tower 210, the pivot axis 213, the shuttle in the boom element 212, the shuttle in the boom element 214, the gripper mounted to the pivot axis 216, the shuttle in the stick element 215, the shuttle in the stick element 217, the shuttle in the stick element 218, the shuttle in the stick element 219, the shuttle in the stick element 220, the gripper mounted about the pivot axis 233, and the brick laying robot 236. For a more detailed description of the brick laying machine, reference is made to the patent specification titled "Brick/Block Laying Machine Incorporated in a Vehicle" which is the subject of international patent application PCT/AU2017/050731, the contents of which are incorporated herein by cross-reference.

[00125] Operation of the brick laying machine will now be discussed. The brick placement database is accessed by control software contained in the control cabinet 282. If a scan of the slab to determine its horizontal variance has not already been carried out, this is now performed, and a trim value array for each brick is determined and loaded as trim data in the brick placement database.

[00126] A brick is taken from a dehacked row of bricks, by the transfer robot, and allocated an identification number as the first brick of the construction according to the brick placement database. The brick is then treated according to the instructions embodied in the brick placement database. As the first brick, it is unlikely that it will require machining or cutting, and if so it will be moved by the transfer robot to the carousel. If the brick requires machining due to the slab scan analysis determining that the slab has a lower elevation at other positions where bricks are to be laid, the brick is moved to the router 247 where it is transferred by a gripper therein and has material machined from the bottom thereof in accordance with the trim value array trim data

stored against the brick in the brick placement database. Otherwise the transfer robot would transfer the brick to the carousel 247. After the machining operation the transfer robot transfers the brick from the router 247 to the carousel 248. After this, the transfer robot is free to return to the row of dehacked bricks, to select the next brick in the sequence as determined by the brick placement database

[00127] The carousel 248 rotates so that its gripper 274 aligns with the transfer robot, and the carousel gripper 274 grips the brick followed by the transfer robot gripper releasing the brick. The carousel 248 rotates to the position of the shuttle and track on the tower 210 (note that this slews with the tower rotating with the boom about the horizontal axis 209). The tower shuttle lowers to the grippers 274, and the brick is transferred to the grippers on the tower shuttle. The tower shuttle can then climb the tower 210 to reach the pivot axis. At this stage the carousel is ready to rotate back to receive the next brick from the transfer robot.

[00128] This process continues, with the transfer robot moving the bricks via the saw 246 and/or router 247, for the cutting and machining each said brick in accordance with said brick placement data allocated against each identified brick in the brick placement database.

[00129] The control software controls the brick laying machine elements to convey each said brick one by one, and apply adhesive and locate each said brick in position on the build, according to said brick placement data for each brick contained in the brick placement database.

[00130] In addition to this the control software builds a handling table identifying for each brick, its identification number which can equate to an identification number in the brick placement database, and for each step between different grippers in the sequence of grippers contained in the brick laying machine, identifies the time and the gripper ID.

[00131] All of these transfers between the programmable brick handling apparatus are logged in the handling table, so that the handling table provides a record of which brick is where and when.

[00132] Consequently, if for any reason the brick laying machine must be paused for any reason, such as shutting down at the end of a working day, it may be restarted, and the correct brick will be laid in the correct position after the restart has occurred.

[00133] Further, if for any reason a brick is damaged during a machining (cutting or routing) operation, it may be discarded at the machining tool, and the handling table can

be updated by reallocating brick identification numbers to the bricks preceding the damaged brick in the supply chain.

[00134] Where damage to an individual brick is not determined until it reaches a position closer to the brick laying gripper, where any said brick already in transit along said handling devices includes no machining data in said brick placement database, said handling table can be updated by reallocating brick identification numbers to the bricks preceding the damaged brick in the supply chain.

[00135] However, where any said brick already in transit along said handling devices includes machining data in said brick placement database, due to the individual bricks being laid in order, it becomes necessary for the control software to run the brick laying machine in reverse, restacking the bricks and storing their restacked position until there are no bricks having associated said machining data in transit along said handling devices, whereupon a replacement brick can be picked from said pallet and processed according to said brick placement database. Thereafter, any restacked bricks are picked up in order, and operation then continues as pre-programmed.

[00136] Referring to figure 28, scanning equipment comprising a camera 350 and a scanning laser 352 scans a brick 354 that has been subject to a machining operation in the router 247 to produce a power outlet recess 356 and part of a cable access channel 357. In the example, the brick 354 has cracked and a chip 358 has come away from the brick. The control software produces a scan file image 360 which is compared with a model file 362 for the brick which is contained in the brick placement database, to determine whether the scan file image 360 differs sufficiently from the model file 362, that the brick 354 should be rejected.

[00137] Both the design software and the control software access a database which is illustrated in figures 29A to 29F. The database can be notionally divided into four sections, namely build data for a house and current machine state, equipment and block transfer logging, vision system logging, and axis/drive and instrument logging. Different elements of the bricklaying machine may access the same data.

[00138] The build data and machine state section of the database stores the current state of the house; including location/state of each block, state of each clamp of the machine, state of the de-hacking bays and slab details. The build data is generated by the computer aided design software, and contains all of the information required by the control software to run the brick laying machine to build a structure. For each house or structure, hereafter referred to as a house for brevity, a record is added to the house

table. This is the identifying data for the particular house being built and includes street address, slab origin in real world GPS coordinates, slab heading and project/client information.

TABLE 1: HOUSE TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
Location	VARCHAR(1000)	The address/lot of a house/structure location
GpsN	REAL	The North GPS location of the house slab origin
GpsE	REAL	The East GPS location of the house slab origin
Direction	REAL	The slab direction, 0-360°
ProjNumber	VARCHAR(1000)	The company project number for the house being built
Customer	VARCHAR(1000)	The customer that the house is being built for
BldCmplt	BOOL	Indicates if the house has completed
BldInProg	BOOL	Indicates if the house has begun

[00139] A house is made up of multiple types of block. Each type is stored in the blocktype table. This contains the block details such as name, manufacturer, ordering details and physical characteristics.

TABLE 2: BLOCKTYPE TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
Maker	VARCHAR(1000)	The company that makes the block
Description	VARCHAR(1000)	The block type and description
OrderCode	VARCHAR(1000)	The order code for the block type
Surface	VARCHAR(1000)	The color and type of block, eg red smooth clay
Long	REAL	The length of the block
Wide	REAL	The width of the block
High	REAL	The height of the block

[00140] A house is made up of many blocks, the details of each block are located in the block table. Each block record references the house that it belongs to as well as its type. As well as the location and rotation of the block on the slab, the block record contains the cut length required and if the spigot is to be removed.

TABLE 3: BLOCK TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
houseID	INT (Relation)	The 'house' that the block belongs to, many 'block' records can reference the same 'house'
blocktypeID	INT (Relation)	The 'blocktype' representing this block, many 'block' records can reference the same 'blocktype'
BlockOrder	INT	The ideal order in the house that the block should be placed
PosX	REAL	The X position of the block location in the house in slab coordinates

PosY	REAL	The Y position of the block location in the house in slab coordinates
PosZ	REAL	The Z position of the block location in the house in slab coordinates
Direction	REAL	The direction of the block in the house in slab coordinates
SawLength	REAL	The length the block must be cut to
RemSpigot	BOOL	Specifies if the spigot needs to be removed

[00141] Although blocks have a BlockOrder specifying ideal lay order, this is not mandatory. Blocks can be laid in a different order, for example, for fault handling. There are however some situations where a block (a) must be laid before another block (b), for example, to prevent the laying arm clamp from colliding with already placed blocks. The blockdependency table allows these situations to be specified.

TABLE 4: BLOCKDEPENDENCY TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
blockID	INT (Relation)	Block (b) that this dependency record specifies, many 'blockdependency' records can reference the same 'block'
dependentblockID	INT (Relation)	Block (a) that this dependency record specifies, many 'blockdependency' records can reference the same 'block'

[00142] For each side of each block requiring routing, a record is placed in the side table. Each record has a link to the block it is associated with. The actual routing information is contained in additional tables.

TABLE 5: SIDE TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
blockID	INT (Relation)	The 'block' that this 'side' belongs to, many 'side' records can reference the same 'block'
SideID	CHAR	Specifies if the side is the front, back, top, underside, left or right side of the block.

[00143] The bore table references the side table, and uses a record for each hole to be drilled into the face of the block. Each hole has a specified position, width and depth. The location is specified relative to the face origin.

TABLE 6: BORE TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
sideID	INT (Relation)	The 'side' that this 'bore' will be performed on, many 'bore' records can reference the same 'side'
HoleXPos	REAL	The bore X position relative to the origin of the block side
HoleYPos	REAL	The bore Y position relative to the origin of the block side
HoleRadius	REAL	The size of the bore to be drilled

HoleDeep	REAL	The depth of the bore to be drilled

[00144] The channel table contains details for each channel that has to be put on a block face. It references the face it is to be machined on as well as the channel depth and width. The channel does not have to be a single straight line but can have corners. The channel path is defined by the channelpoint table. Each point references a chase record and are in local face coordinates. Each chase has 2 or more points making up a path for the channel.

TABLE 7: CHANNEL TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
sideID	INT (Relation)	The 'side' that this 'channel' will be performed on, many 'channel' records can reference the same 'side'
Wide	REAL	The width of the channel
Deep	REAL	The depth of the channel

TABLE 8: CHANNELPOINT TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
channelID	INT (Relation)	The 'channel' that this 'channelpoint' is part of, the 'channel' is made of several 'channelpoints'
ChaseXPos	REAL	The point X position relative to the origin of the block side
ChaseYPos	REAL	The point Y position relative to the origin of the block side

[00145] Parts of the data set are used to generate block routing files for the router on the brick laying machine. A routing file is generated for each block that requires routing and uses the data in the block, side, bore, channel and channelpoint tables. This file is copied to the machine along with the rest of the build data.

[00146] Each block type will have associated pallets that the machine needs defined so it can de-hack it. This data is stored in the tables palletdefinition and palletlayer. The palletdefinition table will simply contain the block type, while the palletlayer table will have one record each for the layers in the pallet. Each layer will specify the number of rows and columns, as well as if the rows run left/right or front/back. Front/back is defined as being parallel with the forklift forks.

TABLE 9: PALLETDEFINITION TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
blocktypeID	INT (Relation)	The 'blocktype' that makes up this pallet, each 'blocktype' can have many 'palletdefinitions'

TABLE 10: PALLETLAYER TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
palletdefinitionID	INT	The 'palletdefinition' that this 'palletlayer' belongs to, each 'palletdefinition' can have multiple 'palletlayers'
LayerNo	INT	The order that the layers are in
NoRows	INT	The lumber of rows in this layer
NoColumns	INT	The number of columns in this layer
Dir	INT	The direction of the layer, 4 directions are possible as the side with the spigot is significant

[00147] Along with the data required to build a structure, the machine state data section of the database contains the information required for the machine to continue building, either at the beginning of a new day, or following an unscheduled shut-down of the H109 controller. When the brick laying machine is restarted after a fault, the data is read back into the controller so that the brick laying machine can continue with minimal operator assistance.

[00148] For each structure that the brick laying machine has in its database, it records if the house has started or completed in the house table.

[00149] A surveypoint table is used to record points built into the slab that can be used to re-align the laser tracker with the slab after the first course of blocks have been laid. Each surveypoint record references the house table that it is associated with.

TABLE 11: SURVEYPOINTS TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
houseID	INT (Relation)	The 'house' that this 'surveypoints' belongs to, many 'surveypoints' records can reference the same 'house'
PosX	REAL	The X position of the point in slab coordinates
PosY	REAL	The Y position of the point in slab coordinates
PosZ	REAL	The Z position of the point in slab coordinates

[00150] For each of the two de-hacker bays 249 and 250, a record is kept in the loadingstate table. The table records the pallet type, current layer and row that is to be de-hacked and the orientation of the pallet in the bay. The table also records if a row of blocks is currently being held in the de-hacker clamp. The primary ID is used to identify the left and right de-hacker data.

TABLE 12: LOADINGSTATE TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	

palletdefinitionID	INT	The pallet type that is loaded in the de hacker bay, this corresponds to the 'palletdefinition' IDX field but there is no database relationship
ThisLayer	INT	The current/next layer to be de-hacked
ThisRow	INT	The current/next row to be de-hacked
Direction	INT	The pallet direction in the bay
ClampPosition	BOOL	Specifies if the de-hacker has hold of a row of blocks

[00151] For each block that has been loaded into the brick laying machine, either from the de-hacker platforms or saw offcut, an entry is made in the blockstate table. This record references the dehackoperation or sawoperation that produced the block. This is separate from the block table used for house data as a physical block may be damaged, and a second blockstate record generated for the second instance of the block in the house. Also, an offcut from the saw requires a blockstate entry, even if it hasn't yet been assigned to the house. The records specify the blocks current location on the H109, record if the cutting, routing and spigot removal have been completed, the current length of the block and the length of block removed from each end. This data is required in addition to the clamp data (described in the next paragraph) as it can track the difference between a block being damaged and discarded, vs being laid on the house. When a block is QC laser scanned, the actual block dimensions will be stored in the table.

[00152] Also associated with the blockstate table are the head flipper and laying arm logs, as these are only stored once per block. These record the start time and operation time of each operation.

TABLE 13: BLOCKSTATE TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
blockID	INT (Relation)	The 'block' that that this 'blockstate' is refereeing to, many 'blockstate' records can reference the same 'block' in the case of block breakage
dehackeroperationID	INT (Relation)	The 'dehackoperation' that produced this 'blockstate', each 'dehackoperation' can have many 'dehackstate' records
sawoperationID	INT (Relation)	The 'sawoperation' that produced this 'blockstate', each 'sawoperation' can have only one 'dehackstate' record
LocID	INT	The current location on the machine that the block is in
CutComp	REAL	Specifies if the block has been cut to length
RouteComp	REAL	Specifies if the block has been routed
SpigotComp	REAL	Specifies if the spigot has been removed
FrontLengthRem	REAL	Specifies the length of block removed from the front (front and back refer to the two block ends)
BackLengthRem	REAL	Specifies the length of block removed from the back (front and back refer to the two block ends)
CurrentLength	REAL	Specifies the current length of the block
Long	REAL	The QC scanned actual length of the block

Wide	REAL	The QC scanned actual width of the block
High	REAL	The QC scanned actual height of the block
FlipLogTime	TIMESTAMP	The start time of the flipper operation
FlipOperationTime	TIME	The time taken for the flipper to complete brick delivery
ArmLogTime	TIMESTAMP	The start time of the arm operation
ArmOperationTime	TIME	The time taken for the arm to complete brick delivery

[00153] To complement the blockstate table, there is a locationstate table to record the state of all clamps on the machine. This table records if the clamp has a block and what its ID is. This is required in addition to the block state data as during block transfers, two clamps can be holding the same block. This data is used to prevent the machine moving two clamps apart when they both have hold of a block as it could cause damage to the machine.

TABLE 14: LOCATION TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
clampID	INT	The ID of the physical clamp on the machine
clampPosition	BOOL	Specifies if the clamp has hold of a block
BlockID	INT	The block type that is loaded in the de hacker bay, this corresponds to the 'block' IDX field but there is no database relationship

[00154] To perform equipment logging, each item of equipment of has its own data structure type in a PLC. Separate buffers for each equipment struct type are used for logs across the brick laying machine. If the tail does not get appended successfully, an alarm will be raised.

[00155] Each time a piece of machine equipment moves or handles a block, the details are recorded. Typically this includes the block ID, what the equipment was doing, performance parameters and any applied corrections from the Vision system.

[00156] The conveyoroperation table records the pallet type that was loaded and operational data. The operational data consists of operation time, the mean torque, the torque standard deviation, the lag error mean and the lag error standard deviation. The operational data will aid the detection of equipment performance changes over time. Conveyor operation records are not attached to a particular house, as the blocks from a single pallet could be used to build adjacent houses without machine pack up in between.

TABLE 15: LOCATION TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
palletdefinitionID	INT (Relation)	Links to the 'palletdefinition' that defines the pallet type loaded by the 'conveyoroperation' record, each

		'palletdefinition' can have many 'conveyoroperation' records
LogTime	TIMESTAMP	The time that the operation began
OperationTime	TIME	The time it took to perform the operation
DehackerID	INT	The de-hacker (eg left or right) that performed the operation
MeanTorque	REAL	The mean (average) torque of the axis during the operation
TorqueDeviation	REAL	The statistical standard deviation of the axis torque during the operation
MeanLagError	REAL	The mean (average) lag error of the axis during the operation
LagErrorDeviation	REAL	The statistical standard deviation of the axis lag error during the operation

[00157] For each row removed from a pallet, a dehack operation is recorded in the dehackoperation table. This includes the results returned from the vision system. That is, the X and Y location as well as the rotation of three detected rows. The three rows are the two edge rows and the centre most row. The centre row is required as the geometry of the bays requires some pallets to be dehacked from the centre rows out. Along with the vision data, the PLC records the pallet layer and what row (of the three returned) was actually picked up. Operational logs are not stored as the de-hackers are 3d motion controlled CNC equipment, and the data cannot easily be reduced to a set of statistical values. Analysis of these drives must be performed on the axis logging data described later. The operation logs will not include the clamp operation as this is handled by the clamp exchange tables described below.

TABLE 16: DEHACKOPERATION TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
conveyoroperatioID	INT (Relation)	The 'conveyoroperation' that that produced the pallet that this 'dehackoperation' is being performed on, each 'conveyoroperation' can have many 'dehackoperation' records
LogTime	TIMESTAMP	The time that the operation began
OperationTime	TIME	The time it took to perform the operation
LayerNo	INT	The layer being dehacked
RowNo	INT	The row (of those returned from the vision system) that was dehacked
PosX1	REAL	Row 1 X position
PosY1	REAL	Row 1 Y position
PosR1	REAL	Row 1 rotation
PosX2	REAL	Row 2 X position
PosY2	REAL	Row 2 Y position
PosR2	REAL	Row 2 rotation
PosX3	REAL	Row 3 X position
PosY3	REAL	Row 3 Y position
PosR3	REAL	Row 3 rotation
RowsDetected	INT	How many rows were detected by the vision system

[00158] For each cut the saw makes, a saw operation is recorded in the sawoperation table. The operation logs will not include the clamp operation as this is handled by the clamp exchange tables described below. The operational logs will hold the table axis operation time, the mean torque, the torque standard deviation and the lag error standard deviation. It will also record the saw torque standard deviation and the velocity error standard deviation.

TABLE 17: LOCATION TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
blockstateID	INT (Relation)	Links to the 'blockstate' that that the 'sawoperation' was performed on, each 'blockstate' can have only one 'sawoperation' record
LogTime	TIMESTAMP	The time that the operation began
OperationTime	TIME	The time it took to perform the operation
MeanTorque	REAL	The mean (average) torque of the saw during the operation

[00159] For each block that the router cuts, a router operation is recorded in the routeroperation table. Operational logs are not stored as the router is a 3d motion controlled CNC equipment, and the data cannot easily be reduced to a set of statistical values. Analysis of these drives must be performed on the axis logging data described later. The logs will not include the clamp operation as this is handled by the clamp exchange tables described below.

TABLE 18: LOCATION TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
blockstateID	INT (Relation)	Links to the 'blockstate' that that the 'routeroperation' was performed on, each 'blockstate' can have only one 'routeroperation' record
LogTime	TIMESTAMP	The time that the operation began
OperationTime	TIME	The time it took to perform the operation

[00160] Each time the transfer robot moves a block from one location to another, a transfer operation log is made in the transferoperation table. This the source and destination clamp locations. The transfer can perform multiple types of block transfer and these require different logs. If the block requires an accurate pick up, the logs include the results returned from the vision system. That is, the X and Y location as well as the rotation of the block. If the transfer is to the carousel, the transfer will include a laser scan. The laser scan logs if the scan resulted in an accurate block match to the expected block. Operational logs are not stored as the transfer is a 3d motion controlled CNC equipment, and the data cannot easily be reduced to a set of statistical values. Analysis

of these drives must be performed on the axis logging data described later. The logs will not include the clamp operation as this is handled by the clamp exchange tables described below.

TABLE 19: TRANSFEROPERATION TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
blockstateID	INT (Relation)	Links to the 'blockstate' that that the 'transferoperation' was performed on, each 'blockstate' can have many 'transferoperation' records
LogTime	TIMESTAMP	The time that the operation began
OperationTime	TIME	The time it took to perform the operation
SourceLoc	INT	The location ID where the transfer picked up the block
DestinationLoc	INT	The location ID where the transfer dropped off the block
PosX	REAL	Vision system correction X component
PosY	REAL	Vision system correction Y component
PosR	REAL	Vision system correction rotation component
BlockMatch	BOOL	If the transfer included a laser scan QC, this field will indicate if the scan matched the model in memory
ClampGrabOp	INT	Refers to the clamp grab operation record, there is no actual database relationship
ClampReleaseOp	INT	Refers to the clamp grab operation record, there is no actual database relationship

[00161] Every time a block is grabbed or released by a clamp (or platform), a clamp operation is logged in the clampoperation and clampexchange tables. These are always in pairs (giving and receiving clamp/platform) and are linked by a clampexchange table. This includes the type of operation (open/close), the clamp side (left/right for individually operated clamp sides) and the operation logs for the clamp. The operational data will aid the detection of equipment performance changes over time. The operational data consists of operation time, the mean torque and the torque standard deviation. Depending on if the clamp is a position or clamp operation, the additional data of velocity error standard deviation or lag error standard deviation is also logged.

TABLE 20: CLAMPOPERATION TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
blockstateID	INT (Relation)	Links to the 'blockstate' that that the 'clampoperation' was performed on, each 'blockstate' can have only one 'clampoperation' record
LogTime	TIMESTAMP	The time that the operation began
OperationTime	TIME	The time it took to perform the operation
IDClamp	INT	The clamp that this record has been logged for
OperationType	BOOL	Is the operation opening or closing
SideClamp	BOOL	Specifies the left or right clamp for two piece clamps
MeanTorque	REAL	The mean (average) torque of the axis during the operation

TorqueDeviation	REAL	The statistical standard deviation of the axis torque during the operation
VeloErrorDeviation	REAL	The statistical standard deviation of the axis velocity error during the operation

TABLE 21: CLAMPEXCHANGE TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
clampoperationdestinationID	INT (Relation)	The 'clampoperation' that is receiving the block, each 'clampoperation' can link to multiple 'clampexchange' records (give and receive)
clampoperationsourceID	INT (Relation)	The 'clampoperation' that is giving the block, each 'clampoperation' can link to multiple 'clampexchange' records (give and receive)

[00162] The vision system logging uses the database frequently. Upon start-up, the vision system will grab the details of every type of block from the database. This information is comprised of properties of the block relevant to the vision system, such as the ideal dimensions of the block, the presence of a spigot and the stacking pattern of the pallet for that block type.

[00163] This information is used frequently through all vision modules once taken from the database, but the system does not query it again, and hence the generic information flow is not documented here.

[00164] However, during operation, the database is queried for more specific information about the blocks. This information relates to the state of the block such as its current location, its measured size, its block type and its bore/channel details.

[00165] Upon completion of each vision analysis, the results are logged into the database. This information is kept for later review if required for any reason, and is detailed enough to trace any single block to all images it appears in.

[00166] The information stored is separated by module, as described in the following sections.

[00167] The dehacker module has a lot more information to log when compared to the other modules, with respect to the database. Every dehacker process has an associated dehackhalcon table with it. As with other modules, one of these records is created for every request the PLC makes.

TABLE 22: DEHACKHALCON TABLE DESCRIPTION

Field	Type	Description

IDX	INT (Primary Key)	
dehackoperationID	INT (RELATION)	Links to the 'dehackoperation' that the 'dehackhalcon' analysis was performed for, each 'dehackoperation' can have many 'dehackhalcon' records
ErrorID	INT	If the analysis could not be completed, the error code returned to the PLC

[00168] Unlike other modules, the dehacker contains three tables of results, the dehackrowshalcon, dehackrowhalcon and the dehackblockhalcon. For every attempted analysis, a dehackrowshalcon record is created, along with a list of block rows, each represented by a single dehackrowhalcon entry. This entry contains the calculated effective centre of the row in 6DOF. For every block row, there exists many blocks, each block is represented by an entry in the dehackblockhalcon table, each containing the 6DOF of the block as well as the type of block. Therefore, every attempt for the dehacker vision module contains one dehackrowshalcon, comprised of many dehackrowhalcon which in turn each have many dehackblockhalcon. The three rows that are selected for PLC return, are indicated in the dehackrowhalcon records.

TABLE 23: DEHACKROWSHALCON TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
dehackhalconID	INT (RELATION)	Links to the 'dehackhalcon' record that the 'dehackrowshalcon' record belongs to, each 'dehackhalcon' record can have many 'dehackrowshalcon' records
LogTime	TIMESTAMP	The time that the operation began

TABLE 24: DEHACKROWHALCON TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
dehackrowshalconID	INT (RELATION)	Links to the 'dehackrowshalcon' record that the 'dehackrowhalcon' record belongs to, each 'dehackrowshalcon' record can have many 'dehackrowhalcon' records
ResultRow	INT	IF this row is a result returned to the PLC, give it the PLC return number (1-3) else this will be 0
PosX	REAL	The X coordinate of the row center
PosY	REAL	The Y coordinate of the row center
PosZ	REAL	The Z coordinate of the row center
PosA	REAL	The A rotation of the row
PosB	REAL	The B rotation of the row
PosC	REAL	The C rotation of the row

TABLE 25: DEHACKBLOCKHALCON TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	

dehackrowhalconID	INT (RELATION)	Links to the 'dehackrowhalcon' record that the 'dehackblockhalcon' record belongs to, each 'dehackrowhalcon' record can have many 'dehackblockhalcon' records
BlockTypeID	INT	The block type for the block being analysed, this corresponds to the 'block' IDX field but there is no database relationship
PosX	REAL	The X coordinate of the block centre
PosY	REAL	The Y coordinate of the block centre
PosZ	REAL	The Z coordinate of the block centre
PosA	REAL	The A rotation of the block
PosB	REAL	The B rotation of the block
PosC	REAL	The C rotation of the block

[00169] Every transfer operation is linked to a single table in the database named transferhalcon. This table represents the link between a PLC operation and the vision system analyses. One of these tables is created for every time the PLC commences a transfer operation to move a block from one location in the base to another location. Each of these transfers will contain one or more transferblockhalcon tables.

[00170] The transferblockhalcon table is the table which contains the actual results of each analysis. The information stored in this table both represent the detected location of the block as well as having the required information to reproduce the analysis if required. Hence, the table contains the results of the analysis as well as the state of the block and the location it is stored in. The results of the analysis being the blocks' X, Y and R locations. The state of the block describes the block at the time of analysis, with regards to its left cut amount and a right cut amount. Finally, the location state comprises of an offset X, Y and R for the analysis location, calculated via offsets provided by the PLC and offsets defined during machine setup/calibration.

TABLE 26: TRANSFERHALCON TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
transferoperationID	INT (RELATION)	Links to the 'transferoperation' record that the 'transferhalcon' record belongs to, each 'transferoperation' record can have many 'transferoperation' records
ErrorID	INT	If the analysis could not be completed, the error code returned to the PLC

TABLE 27: TRANSFEBLOCKRHALCON TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
transferhalconID	INT (RELATION)	Links to the 'transferhalcon' record that the 'transferblockhalcon' record belongs to, each

		'transferhalcon' record can have many 'transferblockhalcon' records
LogTime	TIMESTAMP	The time that the operation began
CutLeft	REAL	The amount of block missing on the left end
CutRight	REAL	The amount of block missing on the right end
OffsetX	REAL	A bay can hold multiple blocks, the represents the X coordinate offset for the block that should be analysed
OffsetY	REAL	A bay can hold multiple blocks, the represents the Y coordinate offset for the block that should be analysed
OffsetR	REAL	A bay can hold multiple blocks, the represents the Rotation offset for the block that should be analysed
PosX	REAL	The actual X coordinate of the block
PosY	REAL	The actual Y coordinate of the block
PosZ	REAL	The actual Rotation of the block

[00171] Every QC laser scanner operation is linked to a single table in the database named laserscanhalcon. This table represents the link between a PLC operation and the vision system analyses. One of these tables is created every time the PLC passes the laser scanner with a block for the carousel, and is populated with one or more result tables. The QC results are very simple. The PLC only receives a pass or fail, with an identifier if failed. As before, there may be a number of analyses for each laserscanhalcon entry, one for each attempt.

TABLE 28: LASERSCANHALCON TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
transferoperationID	INT (RELATION)	Links to the 'transferoperation' record that the 'laserscanhalcon' record belongs to, each 'transferoperation' record can have many 'laserscanhalcon' records
ErrorID	INT	If the analysis could not be completed, the error code returned to the PLC

TABLE 29:LASERSCANBLOCKHALCON TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
laserscanhalconID	INT (RELATION)	Links to the 'laserscanhalcon' record that the 'laserscanblockhalcon' record belongs to, each 'laserscanhalcon' record can have many 'laserscanblockhalcon' records
LogTime	TIMESTAMP	The time that the operation began
BlockMatch	BOOL	Does the scanned block match the 3d model of the block within a pre-determined tolerance
Long	REAL	The scanned length of the brick
Wide	REAL	The scanned width of the brick
High	REAL	The scanned height of the brick

[00172] For every lay head vision operation, there exists a single table named headhalcon. This table represents the link between PLC operation and the vision system analyses. One of these tables is created for every request the PLC requests for the lay head module and is populated with one or more result tables.

[00173] For every headhalcon table there exists at least one table named headblockhalcon. This table contains the actual results of a single vision analysis, which are the 6DOF poses of the blocks. In the case of an intermittent fault, there will be more than one of these headblockhalcon table, one for each attempt.

TABLE 30: HEADHALCON TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
blockstateID	INT (RELATION)	Links to the 'blockstate' record that the 'headhalcon' record belongs to, each 'blockstate' record can have many 'headhalcon' records
ErrorID	INT	If the analysis could not be completed, the error code returned to the PLC

TABLE 31:HEADBLOCKHALCON TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
headhalconID	INT (RELATION)	Links to the 'headhalcon' record that the 'headblockhalcon' record belongs to, each 'headhalcon' record can have many 'headblockhalcon' records
LogTime	TIMESTAMP	The time that the operation began
PosX	REAL	The X component position correction of the actual block location
PosY	REAL	The Y component position correction of the actual block location
PosZ	REAL	The Z component position correction of the actual block location
PosA	REAL	The A component position correction of the actual block location
PosB	REAL	The B component position correction of the actual block location
PosC	REAL	The C component position correction of the actual block location

[00174] For every single instance of any vision result table, the vision system also stores several images used and created during the analysis in a standardised location. These images include the raw image from the image device, as well as various other forms of data generated during analysis. The filenames of the files describe the situation when the request was made.

[00175] For example, the filename "Logs/B1/TM/T8-1-F1P" refers to:

[00176] House 1, Transfer Master, Transfer location 8, PLC operation ID 1, Faulted attempt 1, Processed.

[00177] For the dehacker, the raw image and a processed image are stored for every dehackhalcon entry. The processed image being the image after being rectified into global co-ordinates and has the locations of each block and row marked.

[00178] Dehacker entries are stored with the identifier D#, where # is the ID of the dehacker. The file prefixes match the identifier.

[00179] The transfer stores the raw image from the relevant camera and the processed image after being mapped into the world co-ordinate system, with the detected block outlined.

[00180] These entries are stored with the identifier TM, with the individual entries prefixed T#, where # is the location ID.

[00181] For every laserscanhalcon entry, the raw image from the laser scanner is stored, as well as the point cloud represented in the image, the point cloud of missing points, and the point cloud of extra points. For later reference, the match location of the block in the point cloud is also stored, despite not being relevant to operation.

[00182] QC entries are stored using the identifier QC. With files being prefixed also with QC and suffixed with the content type, where M refers to missing and E refers to extra. Example: QC-2, QC-4-F2M or QC-6-F1E. In theory, there should not be a M or E file for a perfect block.

[00183] For the layhead, all raw images from the camera array as well as the calculated point cloud from the images are stored for every headhalcon entry.

[00184] Layhead entries are stored using the identifier LH, with the numerous images prefixed L1 to L6.

[00185] All axis and instruments have normal logging of operational data. As a drive is operating it will check the following parameters:

- Lag between setpoint and actual position of drive (PositionLag)
- Velocity of drive
- Acceleration of drive
- Torque of drive
- Distance travelled by drive

If any of these change by a significant amount (amount is configurable) an event will be triggered to create a log in the database. Regardless of the trigger for logging, the following variables will be stored:

TABLE 32: DRIVE TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
DriveID	INT	unique ID of the drive
LogTime	TIMESTAMP	The time that the log was generated
TotalDistance	BIGINT	Total distance travelled by drive at this point in time
ExecutePosition	BOOL	Drive is being commanded to move to the set point
PositionSP	REAL	Drive position set point from the controller
Position	REAL	Drive position at time of logging
PositionLag	REAL	Drive position lag (difference between setpoint and actual position) at time of logging
Velocity	REAL	Drive velocity at time of logging
Acceleration	REAL	Drive acceleration at time of logging
Torque	REAL	Drive torque at time of logging

[00186] This data will be stored within a custom-made 'drive-log' structure. This will then be appended as a tail to a buffer of axis log structures. The buffer will be used for all axis logs across the machine. If the tail does not get appended successfully, an alarm will be raised.

[00187] A drive that is defined as a DriveReference will store the above variables, as well as the position, velocity and acceleration relative to the parent that it is referencing. The data will be stored within a custom-made 'DriveReference-log' structure. The buffer will be used for all DriveReference logs across the machine. If the tail does not get appended successfully, an alarm will be raised.

TABLE 33: DRIVEREFERENCE TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
DriveID	INT	unique ID of the drive
LogTime	TIMESTAMP	The time that the log was generated
TotalDistance	BIGINT	Total distance travelled by drive at this point in time
ExecutePosition	BOOL	Drive is being commanded to move to the set point
PositionSP	REAL	Drive position set point from the controller
Position	REAL	Drive position at time of logging
PositionLag	REAL	Drive position lag (difference between setpoint and actual position) at time of logging
Velocity	REAL	Drive velocity at time of logging
Acceleration	REAL	Drive acceleration at time of logging
Torque	REAL	Drive torque at time of logging
ExecutePositionRef	REAL	Drive is being commanded to move to the referenced position SP

PositionRefSP	REAL	Drive referenced position SP from the controller
PositionRef	REAL	Drive referenced position at time of logging
VelocityRef	REAL	Drive referenced velocity at time of logging
AccelerationRef	REAL	Drive referenced acceleration at time of logging

[00188] A drive that is defined as DriveClamp will store the same variables as a DriveReference, as well as total distance the clamp apparatus itself has travelled. It also records the clamp operation state (what the clamp is doing). The data will be stored within a custom-made 'DriveClamp-log' structure. The buffer will be used for all DriveClamp logs across the machine. If the tail does not get appended successfully, an alarm will be raised.

TABLE 34: DRIVEREFERENCE TABLE DESCRIPTION

Field	Type	Description
IDX	INT (Primary Key)	
DriveID	INT	unique ID of the drive
LogTime	TIMESTAMP	The time that the log was generated
TotalDistance	BIGINT	Total distance travelled by drive at this point in time
ExecutePosition	BOOL	Drive is being commanded to move to the set point
PositionSP	REAL	Drive position set point from the controller
Position	REAL	Drive position at time of logging
PositionLag	REAL	Drive position lag (difference between setpoint and actual position) at time of logging
Velocity	REAL	Drive velocity at time of logging
Acceleration	REAL	Drive acceleration at time of logging
Torque	REAL	Drive torque at time of logging
ExecutePositionRef	REAL	Drive is being commanded to move to the referenced position SP
PositionRefSP	REAL	Drive referenced position SP from the controller
PositionRef	REAL	Drive referenced position at time of logging
VelocityRef	REAL	Drive referenced velocity at time of logging
AccelerationRef	REAL	Drive referenced acceleration at time of logging
ClampTotalDistance	BIGINT	Total distance travelled by the clamp at this point in time
OperationState	INT (ENUM)	What is the clamp doing: 0 – no command 1 – opening 2 – open 3 – closing 4 – closed

[00189] The design and control software of the invention provides a complete solution from design through to completed build, of any structure that may be built from bricks or blocks using a brick laying machine.

The Claims Defining the Invention are as Follows

1. Computer aided design software for designing a brick construction, said computer aided design software having a user interface allowing a user to input building plan data, said computer aided design software generating from said building plan data, data representative of a scale top-plan view of walls with predetermined known wall length dimensions, said computer aided design software generating a virtual extrusion of length commensurate with the wall height;
said computer aided design software determining brick by brick placement data for said walls, including the identification of the position and orientation in three dimensions of each brick relative to a point of origin, including determining brick stepping and brick nesting at wall intersections, and having regard to a predetermined minimum cut brick length - determination of cut length data for individual bricks to be cut to length in order to meet the required dimensions of wall length;
said computer aided design software compiling a brick placement database including brick type, position and orientation in three dimensions of each brick relative to said point of origin, and cut length data for each brick identified as to be cut to length.

2. Computer aided design software as claimed in claim 1 having a first table containing one or more user selectable settings allowing stock brick type and size parameters to be stored and selected for walls.

3. Computer aided design software as claimed in claim 1 or 2 wherein said building plan data is representative of a scale top-plan view of external walls and internal walls.

4. Computer aided design software as claimed in any one of the preceding claims wherein said computer aided design software generates footing data and/or concrete pad data including the dimensions, position and orientation of the footings and/or concrete pad relative to a point of origin, relative height off-set between the bottom of the external walls and optionally the bottom of the internal walls and between different sections of internal walls and optionally determines the required volume of concrete to form the pad.

5. Computer aided design software as claimed in any one of the preceding claims wherein said computer aided design software allows user input of and storing

positioning data for voids and/or apertures in said extrusion, said voids and/or apertures being of dimensions commensurate with the height and width of doors and windows to be fitted in the completed building.

6. Computer aided design software as claimed in any one of the preceding claims wherein said computer aided design software allows user input of and storing services positioning data for at least one of plumbing, electrical and ICT (Information and Communication Technology) and sound and vision cabling and connection points in said external walls and in said internal walls; said computer aided design software generating positioning data for channels in said walls to carry services and recesses in said walls for said connection points, said computer aided design software generating machining data for the location of recesses and cavities to be machined in individual bricks according to the positioning data of said channels and recesses; and storing said machining data in said brick placement database.

7. Computer aided design software as claimed in any one of the preceding claims adapted to determine the order that each brick is to be laid, and creates in said brick placement database, an index number allocated to each brick, to identify the brick laying order.

8. Computer aided design software as claimed in any one of the preceding claims wherein the reference position on each brick is the centroid of all dimensions of each brick, trimmed or whole, but for simplicity not taking into account any routed cut-outs or recesses.

9. Computer aided design software as claimed in any one of claims 4 to 8 wherein the relative height off-set between the bottom of the external walls and the bottom of the internal walls may be reflected in different z values for the first course of bricks of the external course and the internal course.

10. Computer aided design software as claimed in any one of the preceding claims wherein different spacings between adjacent bricks and thicknesses of adhesive or mortar between adjacent bricks are accounted for and adjacent brick spacing A and brick base spacing B are stored in the brick placement database.

11. Computer aided design software as claimed in any one of the preceding claims wherein for cut bricks the design software exports the coordinates of the cut to said database.

12. Computer aided design software as claimed in any one of claims 6 to 11 wherein for routed bricks the design software exports the coordinates of the rout to said database.

13. Computer aided design software as claimed in any one of the preceding claims wherein the design software brick placement database includes at least one trim data field to store a trim value or a trim value array for each brick.

14. Computer aided design software as claimed in any one of claims 6 to 13 having a first library in which data pertaining to one or more building plans in the form of said data sets are stored.

15. Computer aided design software as claimed in claim 14 having a second library in which data pertaining to one or more pre-defined rooms are stored.

16. Computer aided design software as claimed in claim 15 having a third library in which data pertaining to a plurality of doors and windows are stored, corresponding to data for stock items used in determining said positional data based on selected doors or windows.

17. Computer aided design software as claimed in any one of claims 6 to 16 wherein said computer aided design software treats walls of a structure to be built as segments extending between intersections of brickwork, where each segment has a course segment extending between intersections of brickwork and window and door opening edges; where any said course segment has a length s, where:

$$s = n.(b + A) + r + A + p.e + p.A + q.f + q.A - A$$

where A is the A value or gap),

b is the stock brick length,

e and f are the end overlap at a brick wall intersection,

p may be 1 (indicating e is equal to the width of a brick at the intersecting wall, or zero (abutting),

q may be 1 (indicating f is equal to the width of a brick at the intersecting wall, or zero (abutting), r is the remainder which may be zero or greater than or equal to 0.2 b, preferably 0.25 b, and if this is true, a single brick is cut to length r to complete the course segment;

and if r is less than 0.2 b, preferably 0.25 b,

$$s = (n - 1).(b + A) + 2r + 2A + p.e + p.A + p.f + q.A - A$$

where r is the length that two bricks are cut to, to locate within and complete a course segment having n-1 bricks.

18. Computer aided design software as claimed in claim 17 wherein a next course of said bricks are arranged in order to achieve preferred stepping, where a said course segment includes two bricks of length r, the immediately overlying course segment includes a single brick of length r balanced on the join between the two bricks of length r, with two bricks cut to length of c=(b+r)/2, located on either side thereof, with bricks of stock brick length b continuing away from at least one of said two bricks cut to length of c, and the course segment length can be described by the following:

$$s = (n - 2).(b + A) + r + A + 2(c+A) + p.e + p.A + p.f + q.A - A$$

19. Control software to control a brick laying machine to build a building or other structure of brick construction, said control software accessing a brick placement database compiled by computer aided design software as claimed in any one of the preceding claims, said control software controlling said machine to cut and optionally to machine each said brick in accordance with data stored in said brick placement database, and controlling said machine to convey each said brick one by one, and apply adhesive and locate each said brick according to data stored in said brick placement database.

20. Control software to control a brick laying machine to build a building or other structure of brick construction, said control software accessing a brick placement database including brick type, position and orientation in three dimensions of each brick relative to a point of origin, cut length data for each brick identified as to be cut to length, and machining data for each brick including a trim value or a trim value array for each brick being trim data derived from data received from a scanner located at a surveying position to measure the relative surface height of a surface extent where bricks are to be laid; said control software correcting for the difference in height of the surface extent for each brick, from the lowest point and

the highest point for each course of bricks and determining from said trim data the amount to be machined from a horizontal face of each brick so that the top of each course of bricks is level when laid; said control software controlling said machine to cut and machine each said brick in accordance with data stored in said brick placement database, said control software controlling said machine to convey each said brick one by one, and apply adhesive and locate each said brick according to data stored in said brick placement database.

21. Control software to control a brick laying machine to build a building or other structure of brick construction, said control software accessing a brick placement database including brick type, position and orientation in three dimensions of each brick relative to a point of origin, cut length data for each brick identified as to be cut to length, and machining data for each brick; said control software including a scanner interface to receive data from a scanner located at a surveying position to measure the relative surface height of a surface extent where bricks are to be laid, storing the height of the surface for at least one location for each brick at which each brick is to be located according to said brick placement database; said control software correcting for the difference in positioning of said surveying position and said point of origin and determining the difference in height of the surface for said at least one location for each brick, from the lowest point and the highest point for each course of bricks and storing the difference in said brick placement database as trim data in the form of a trim value or trim value array for each said brick corresponding with the amount to be machined from a horizontal face of each brick so that the top of each course of bricks is level when laid; said control software controlling said machine to cut and machine each said brick in accordance with data stored in said brick placement database, said control software controlling said machine to convey each said brick one by one, and apply adhesive and locate each said brick according to data stored in said brick placement database.

22. Control software as claimed in any one of claims 19 to 21 wherein said control software determines the order that each brick is to be laid, and creates an index number allocated to each brick, to identify the brick laying order, and enters the index number into the brick placement database.

23. Control software as claimed in any one of claims 19 to 22 wherein said control software includes a library of handling identifiers which each identify a unique handling device within said brick laying machine, and said control software produces a handling table identifying individual bricks and associating individual

bricks with a particular handling device at a particular time, and updating said handling table as individual brick progress from handling device to handling device with the elapsing of time.

24. Control software as claimed in any one of claims 19 to 23 wherein said control software calculates corrected x y z position and orientation data relative to said point of origin for each brick recorded in said brick placement database, based on the difference between the location of the point of origin and the surveying position, and records the corrected x y z position and orientation data relative to said surveying position, for use in controlling said brick laying machine.

25. Control software as claimed in any one of claims 19 to 24 wherein said trim data is measured and stored as a trim value array for multiple locations for each said brick, so that said machine may machine said brick to compensate for localised footing or pad height excesses.

26. A method of building a structure, comprising steps of determining the size of brick to be utilised; creating a scale top-plan view of walls with predetermined known wall length dimensions; determining brick by brick placement data for said walls, including identification of the position and orientation in three dimensions of each brick identification of the position in three dimensions and orientation of individual bricks to be cut to length in order to meet the required dimensions of wall length, and determining the order that each brick is to be laid, and storing this data in a brick placement database; measuring the relative surface height of a surface extent where bricks are to be laid, recording the height of footings for at least one location at which each brick is to be located according to said brick placement data, determining the difference in height of the footings for at least said one location for each brick, from the lowest point to the highest point for at least the first course of bricks and storing the difference from the lowest point as trim data in the form of a trim value or trim value array for said at least one location for each said each brick corresponding with the amount to be machined from a horizontal face of each brick so that the top of at least the first course is levelled when laid, the trim data being stored with said brick placement data; cutting each said brick in accordance with said brick placement data, conveying each said brick one by one, and applying adhesive and locating each said brick according to said brick placement data.

27. A method of building a building or other structure, comprising steps of determining the size of brick to be utilised for external walls and the size of brick to be utilised for internal walls; creating a scale top-plan view of external walls and internal walls

with predetermined known wall length dimensions, determining footing and/or concrete pad data including the dimensions of the footings and/or concrete pad, relative height off-set between the bottom of the external walls and the bottom of the internal walls and between different sections of internal walls and optionally determining the required volume of concrete to form the pad; planning the configuration of the walls from the footings and/or pad up to the tops of the walls including positional determination of voids of dimensions commensurate with the height and width of doors and windows to be fitted, and positioning data for channels and recesses (chasing) for services of plumbing, electrical and ICT and sound and vision cabling and connection points in said external walls and in said internal walls; determining brick by brick placement data for said external walls and said internal walls, including identification of the position and orientation in three dimensions of each brick relative to the footings and/or concrete pad, identification of the position in three dimensions of individual bricks to be cut to length in order to meet the required dimensions of wall length, void size and aperture size, and generating machining data for the position of recesses and cavities to be machined in individual bricks according to the positioning data of said channels and recesses, and determining the order that each brick is to be laid, and storing this data in a brick placement database; measuring the relative surface height of a surface extent where bricks are to be laid , recording the height of the footings and/or a concrete pad for at least one location at which each brick is to be located according to brick placement data as described above, determining the difference in height of the pad for at least said one location for each brick, from the lowest point to the highest point for at least the first course of bricks and storing the difference from the lowest point as trim data in the form of a trim value or trim value array for said at least one location for each said each brick corresponding with the amount to be machined from a horizontal face of each brick so that the top of at least the first course is level when laid, the trim data being stored with said brick placement data; cutting and machining each said brick in accordance with said brick placement data, conveying each said brick one by one, and applying adhesive and locating each said brick according to said brick placement data.

28. The method as claimed in claim 26 or 27 wherein the building or structure may be built course by course until completed to the required height.

29. The method as claimed in claim 20 wherein the first course of each course of bricks is machined to reduce their height as necessary, according to the data from

measuring the relative surface height of footings and/or a concrete pad so the tops of the first course are level.

30. The method of any one of claims 19 to 21 wherein said trim data is measured and stored as a trim value array for multiple locations for each said brick, so that each said brick is machined to compensate for localised footing or pad height excesses.

31. The method of any one of claims 19 to 21 wherein position and orientation data for each brick includes x and y horizontal dimensions relative to a reference point, with reference to a position on each brick, z vertical dimension, and Ø angle relative to magnetic north or other direction.

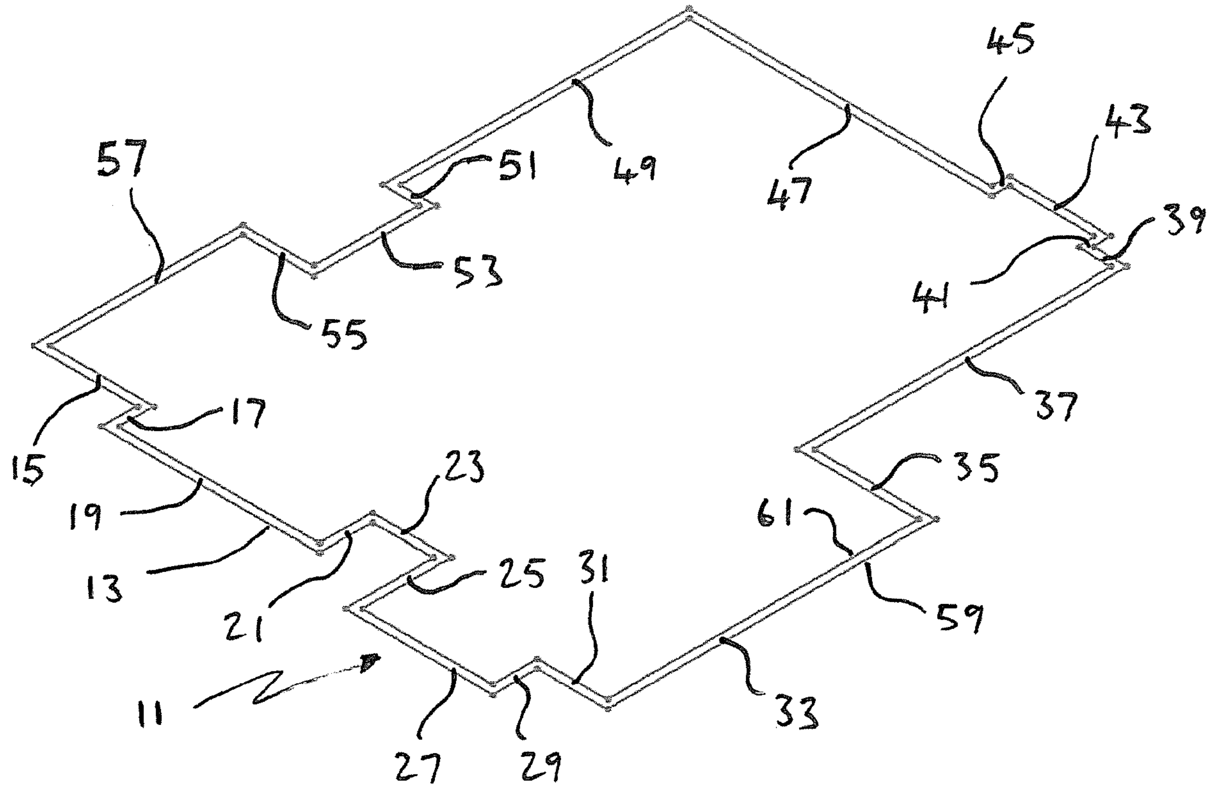

FIGURE 1

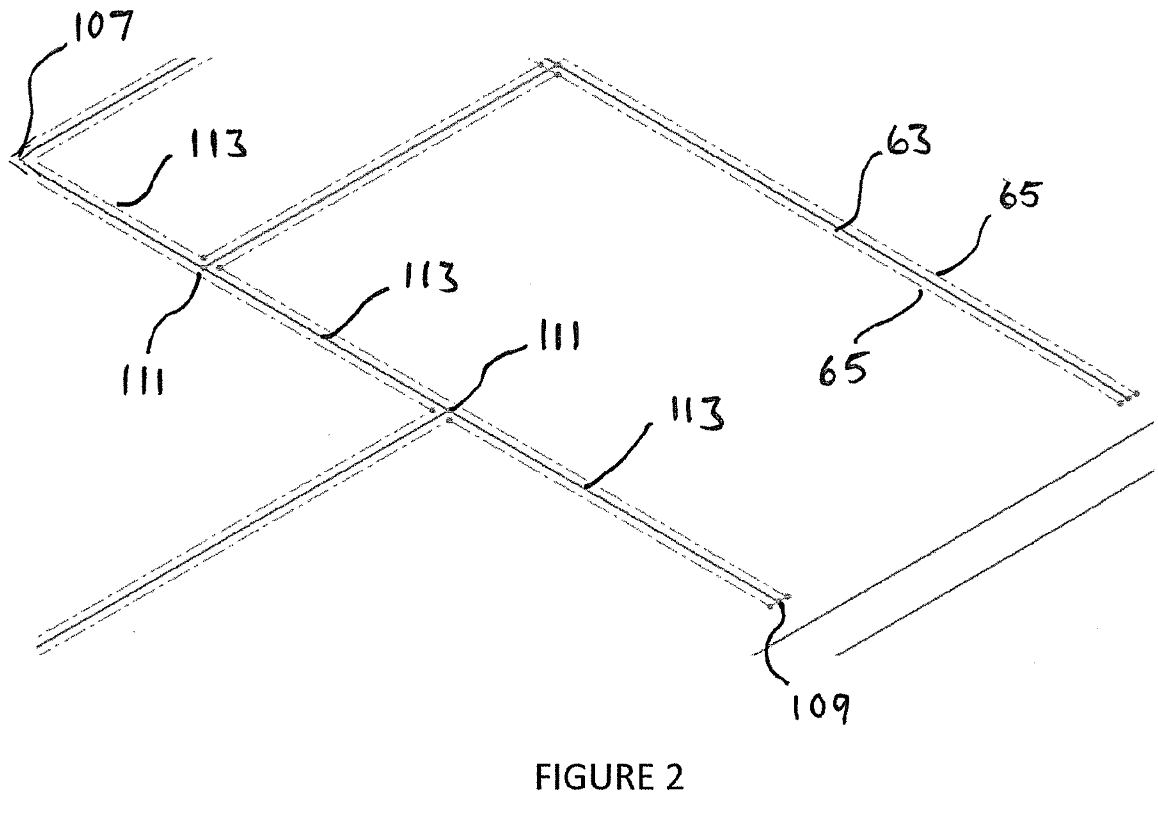

FIGURE 2

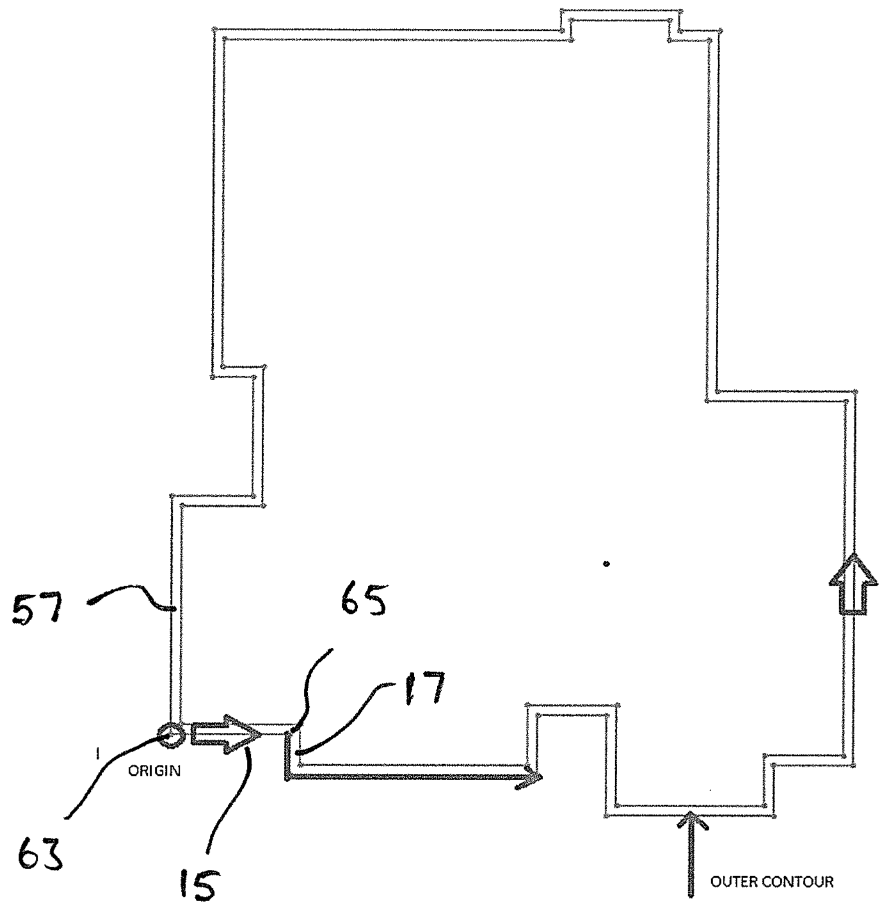

FIGURE 3

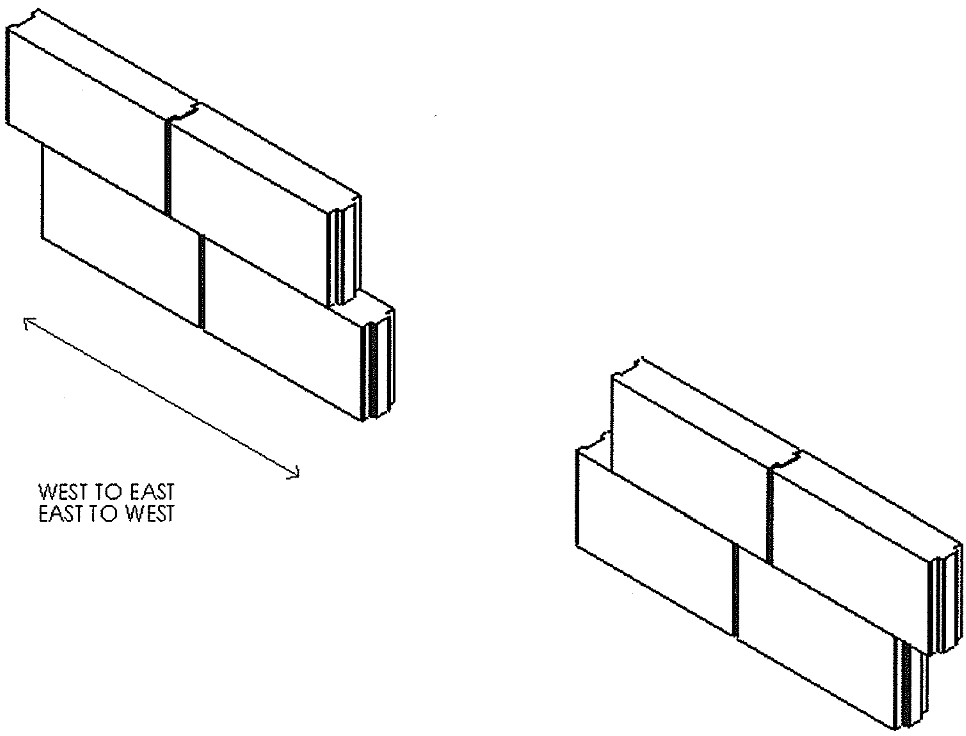

FIGURE 13

1	External Wall Library File	Text	BRICKS\BR000004\BR000004-001.SLDPRT	BRICKS\BR000004\BR000004-001.SLDPRT
2	External Wall Brick Depth	Text	250	250
3	External Wall Brick Length	Text	500	500
4	External Wall Brick Height	Text	258	258
5	External Wall Vertical Brick Gap	Text	0	0
6	External Wall Horizontal Brick Gap	Text	0	0
7	External Inside Top Wall Library File	Text	BRICKS\BR000007\BR000007-001.SLDPRT	BRICKS\BR000007\BR000007-001.SLDPRT
8	External Inside Top Wall Brick Depth	Text	125	125
9	External Inside Top Wall Brick Length	Text	500	500
10	External Inside Top Wall Brick Height	Text	258	258
11	External Inside Top Wall Vertical Brick Gap	Text	0	0
12	External Inside Top Wall Horizontal Brick Gap	Text	0	0
13	Internal Wall Library File	Text	BRICKS\BR000006\BR000006-001.SLDPRT	BRICKS\BR000006\BR000006-001.SLDPRT
14	Internal Wall Brick Depth	Text	125	125
15	Internal Wall Brick Length	Text	500	500
16	Internal Wall Brick Height	Text	258	258
17	Internal Wall Vertical Brick Gap	Text	0	0
18	Internal Wall Horizontal Brick Gap	Text	0	0
19	Internal Wall Rebate	Text	115	115
20	1st Floor Slab Type	Text	LevelSlab	LevelSlab

FIGURE 4

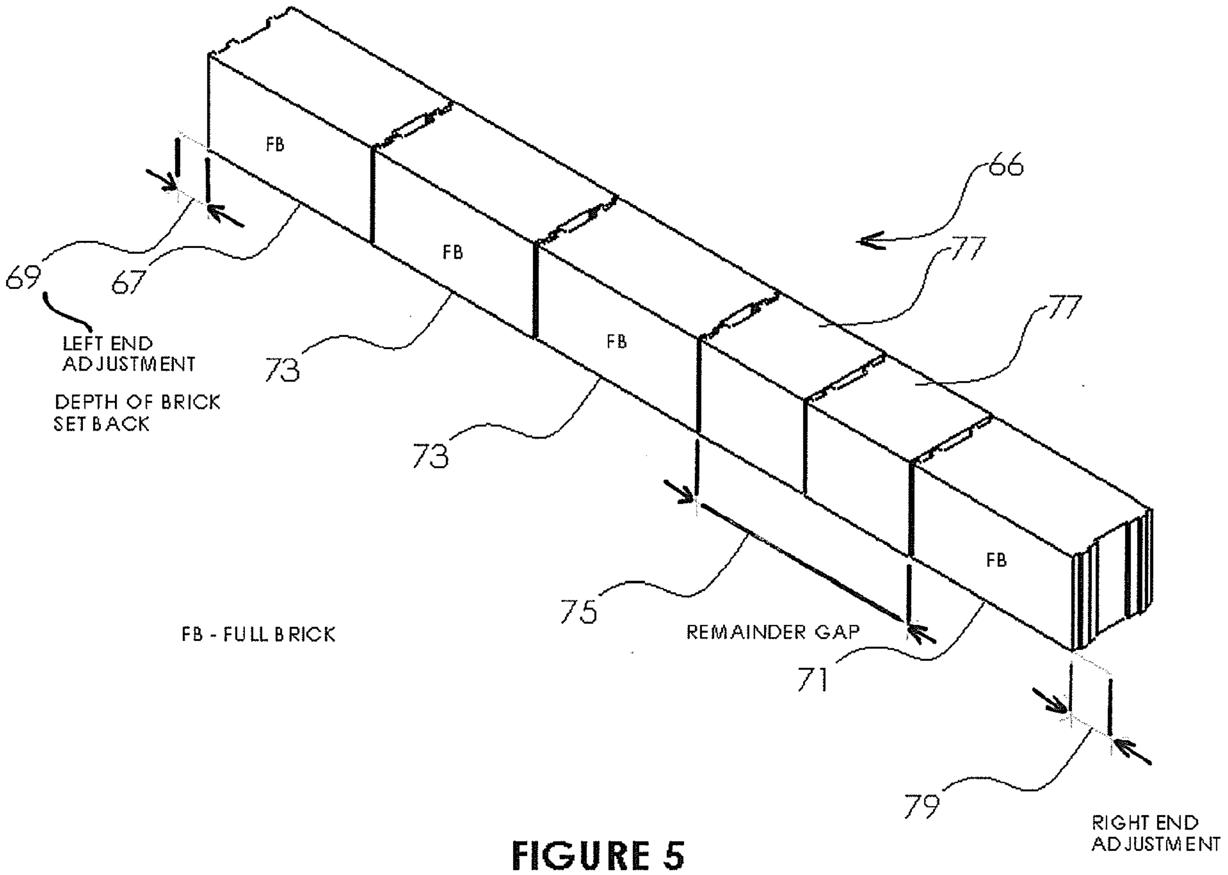

FIGURE 5

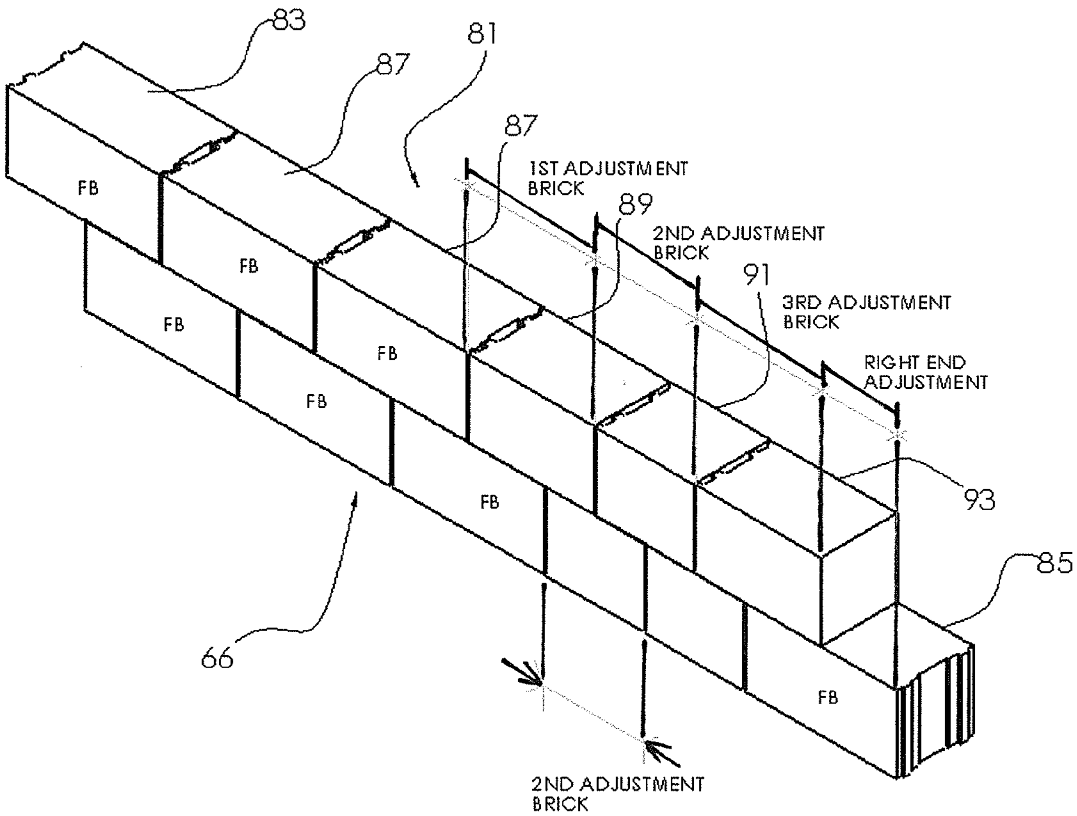

FIGURE 6

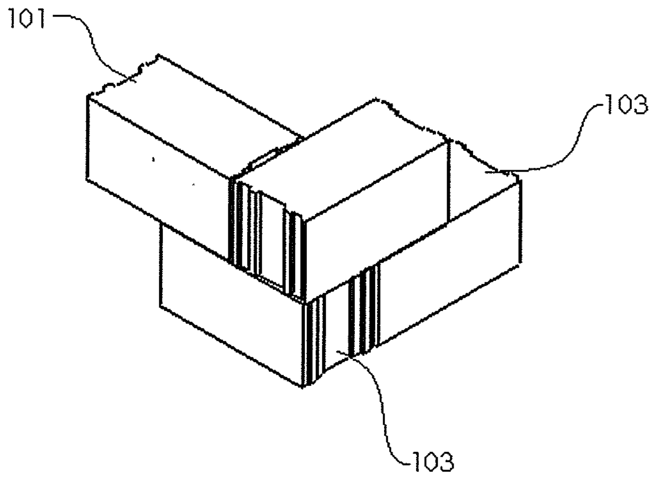

FIGURE 7

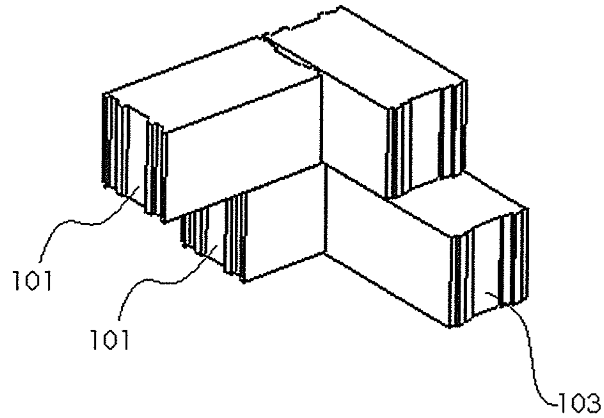

FIGURE 8

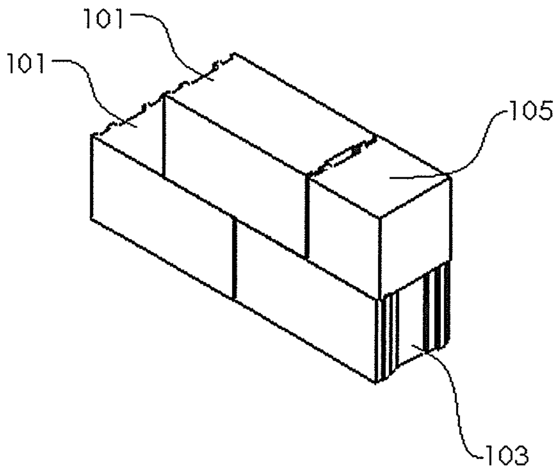

FIGURE 9

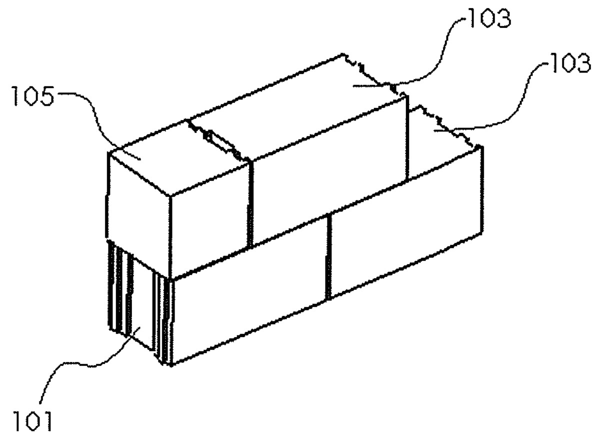

FIGURE 10

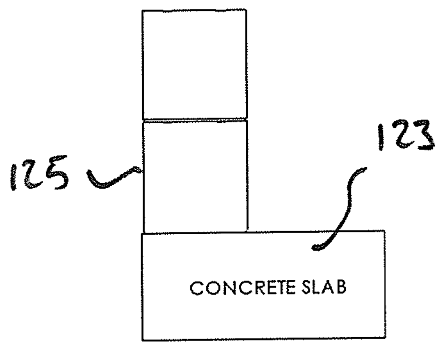

FIGURE 22

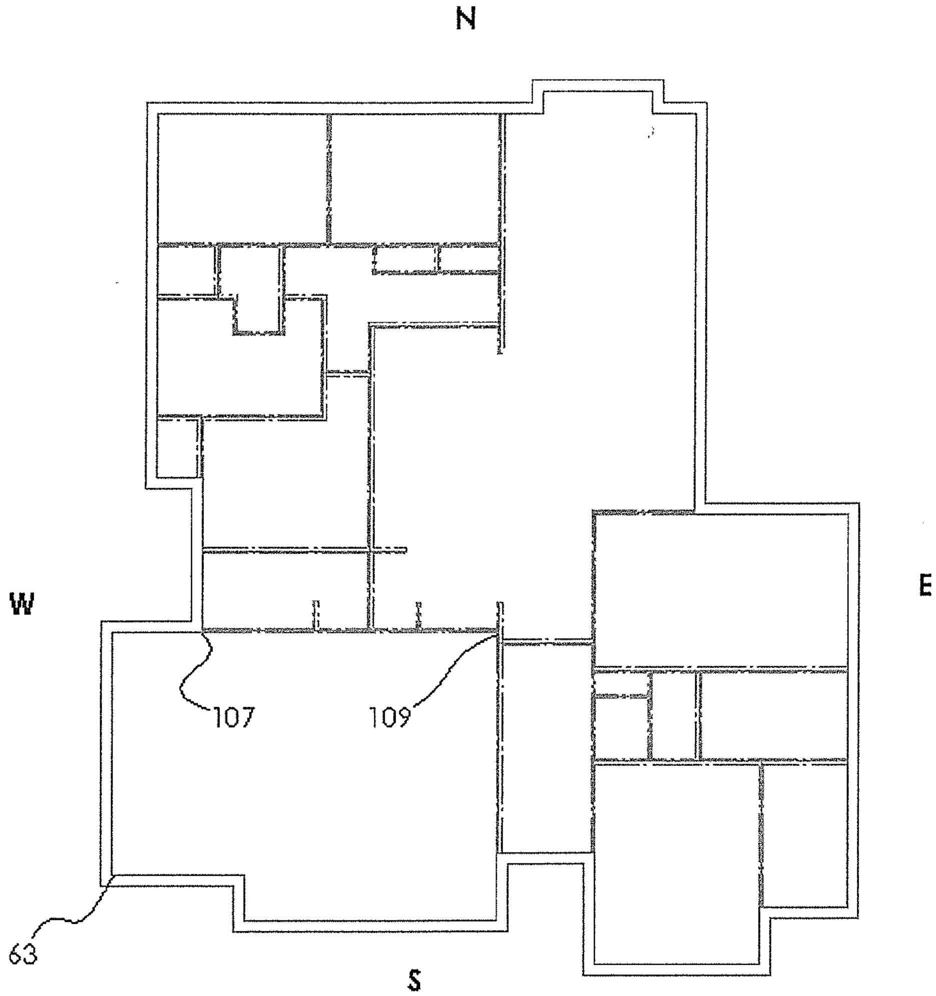

FIGURE 11

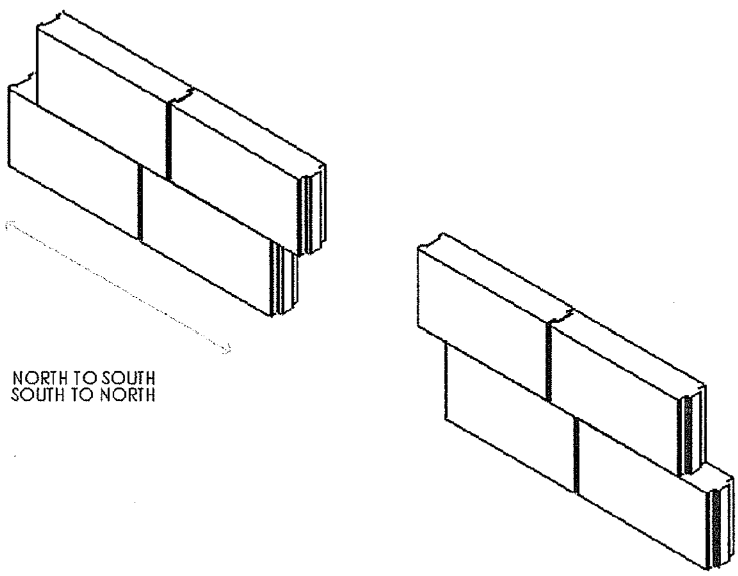

FIGURE 12

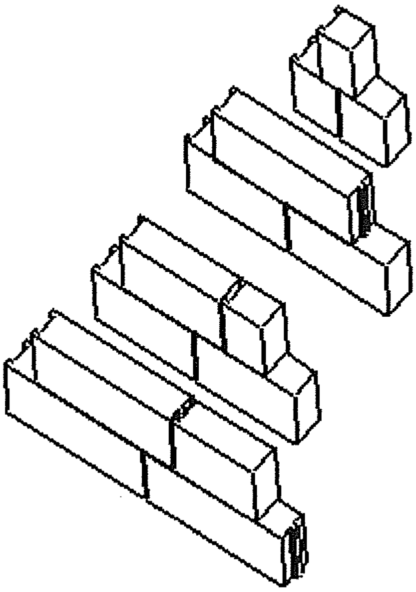

FIGURE 14

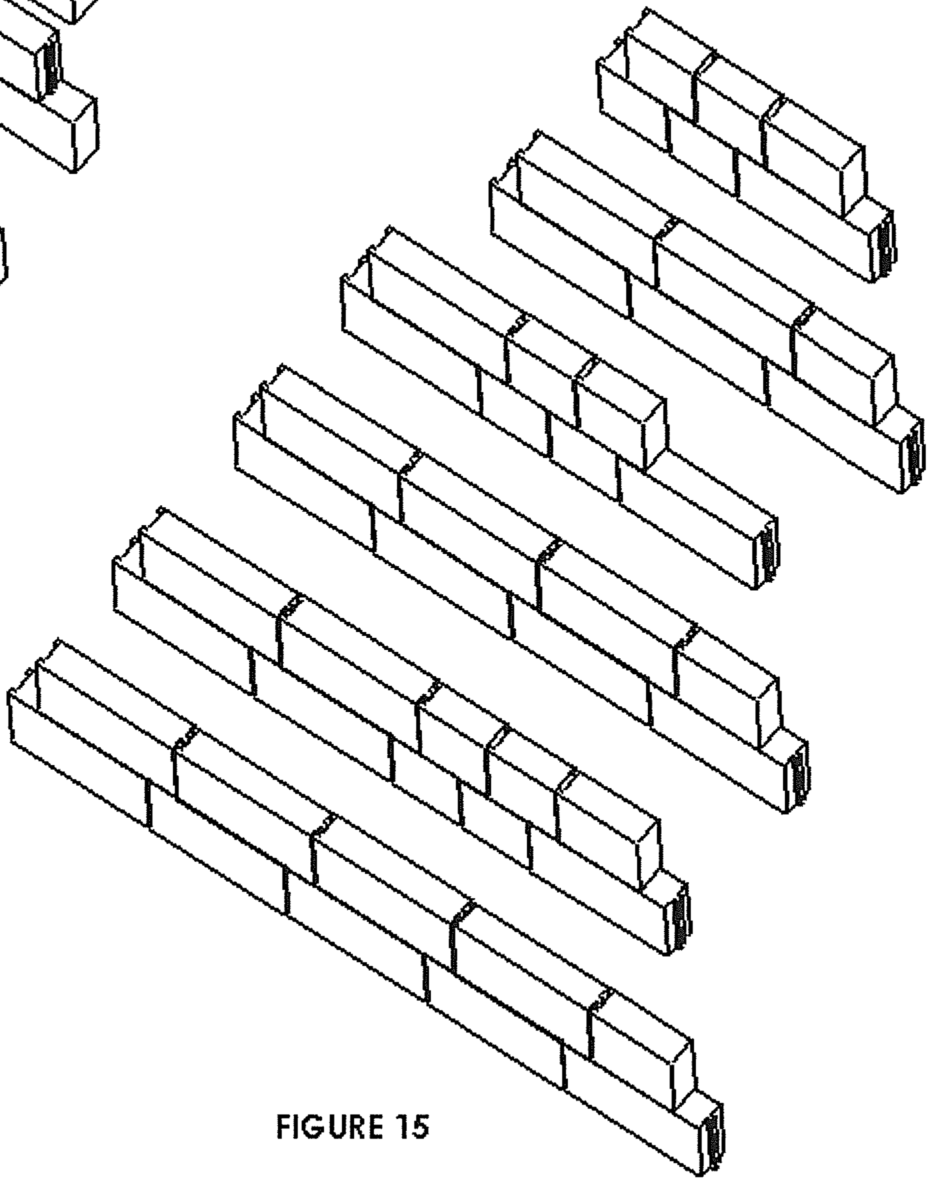

FIGURE 15

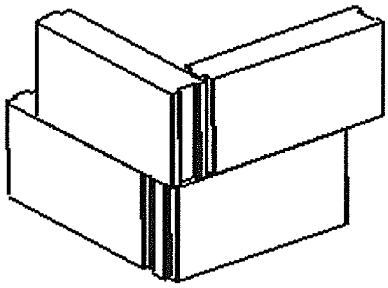

FIGURE 16

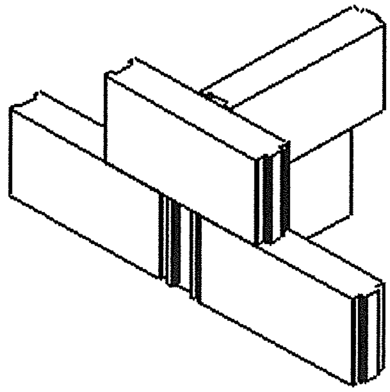

FIGURE 17

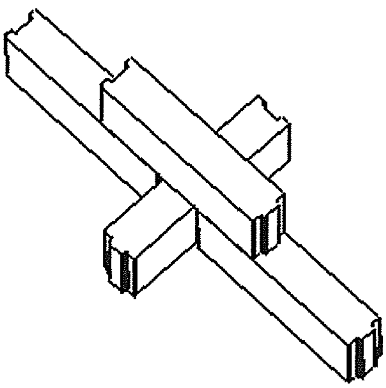

FIGURE 18

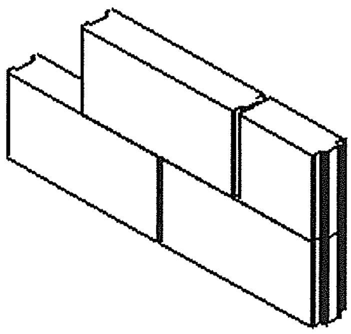

FIGURE 19

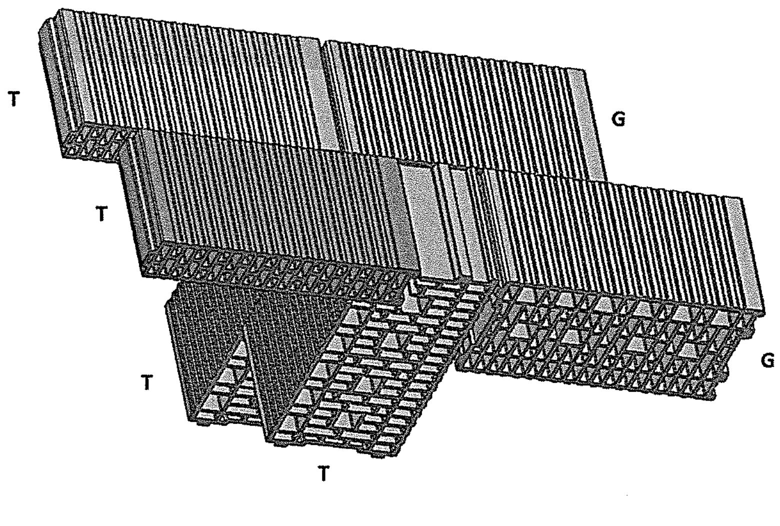

FIGURE 20

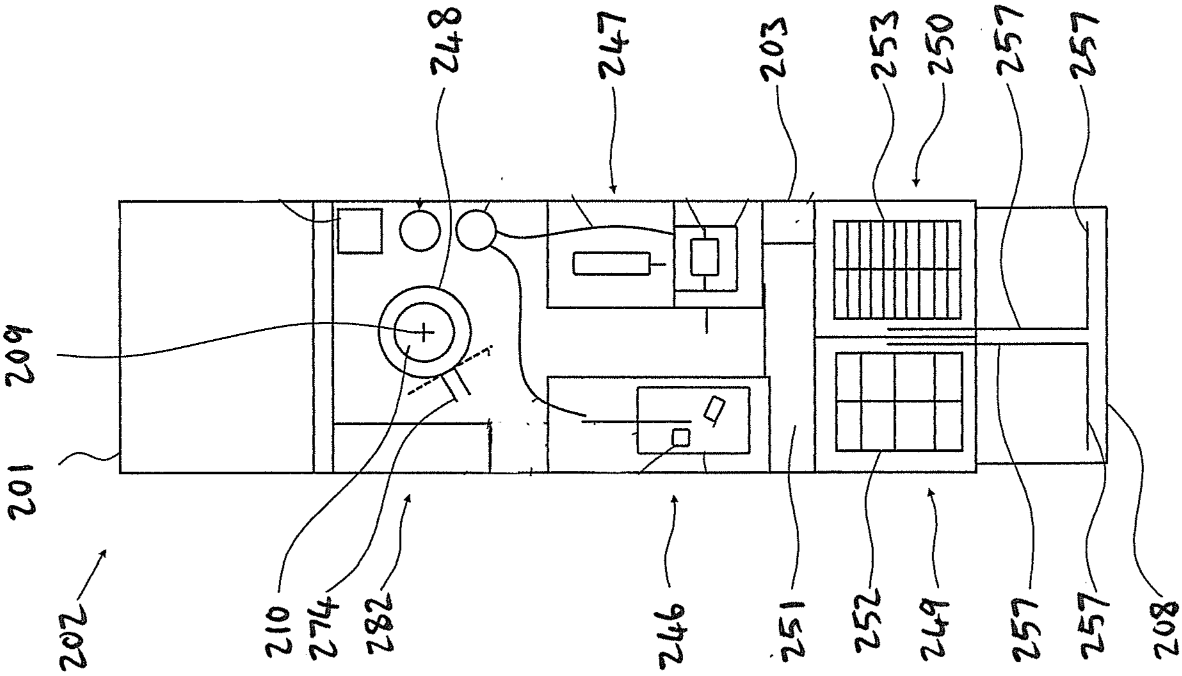

FIGURE 26

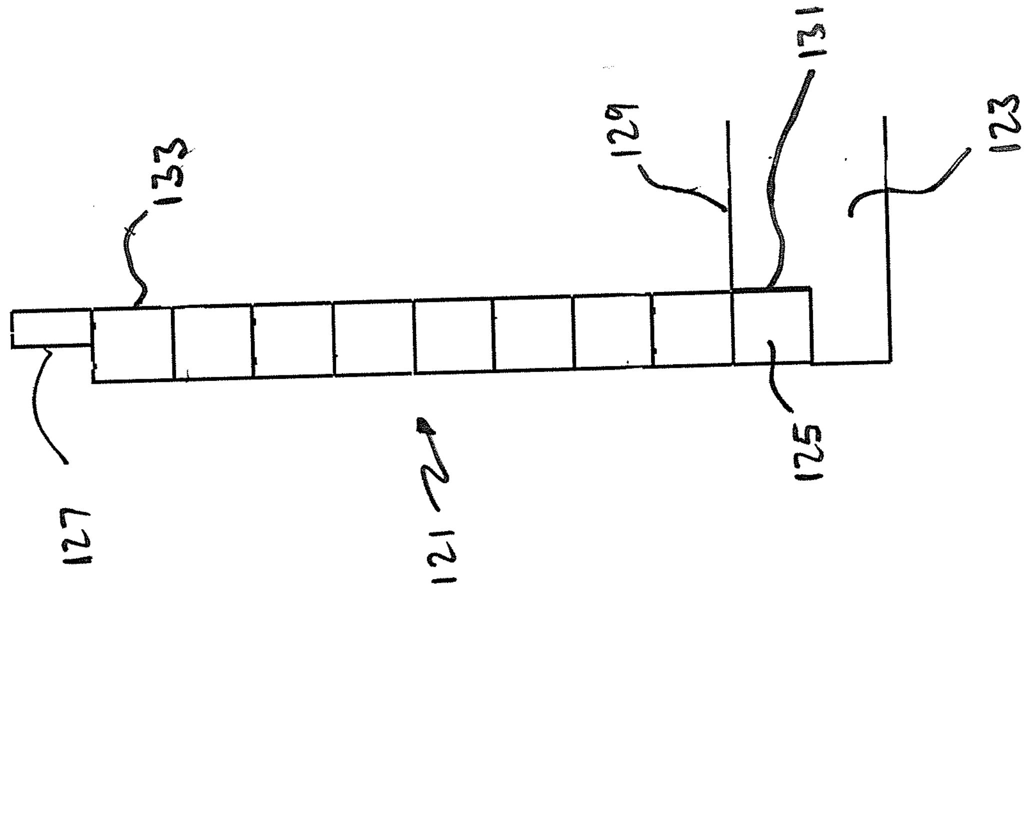

FIGURE 21

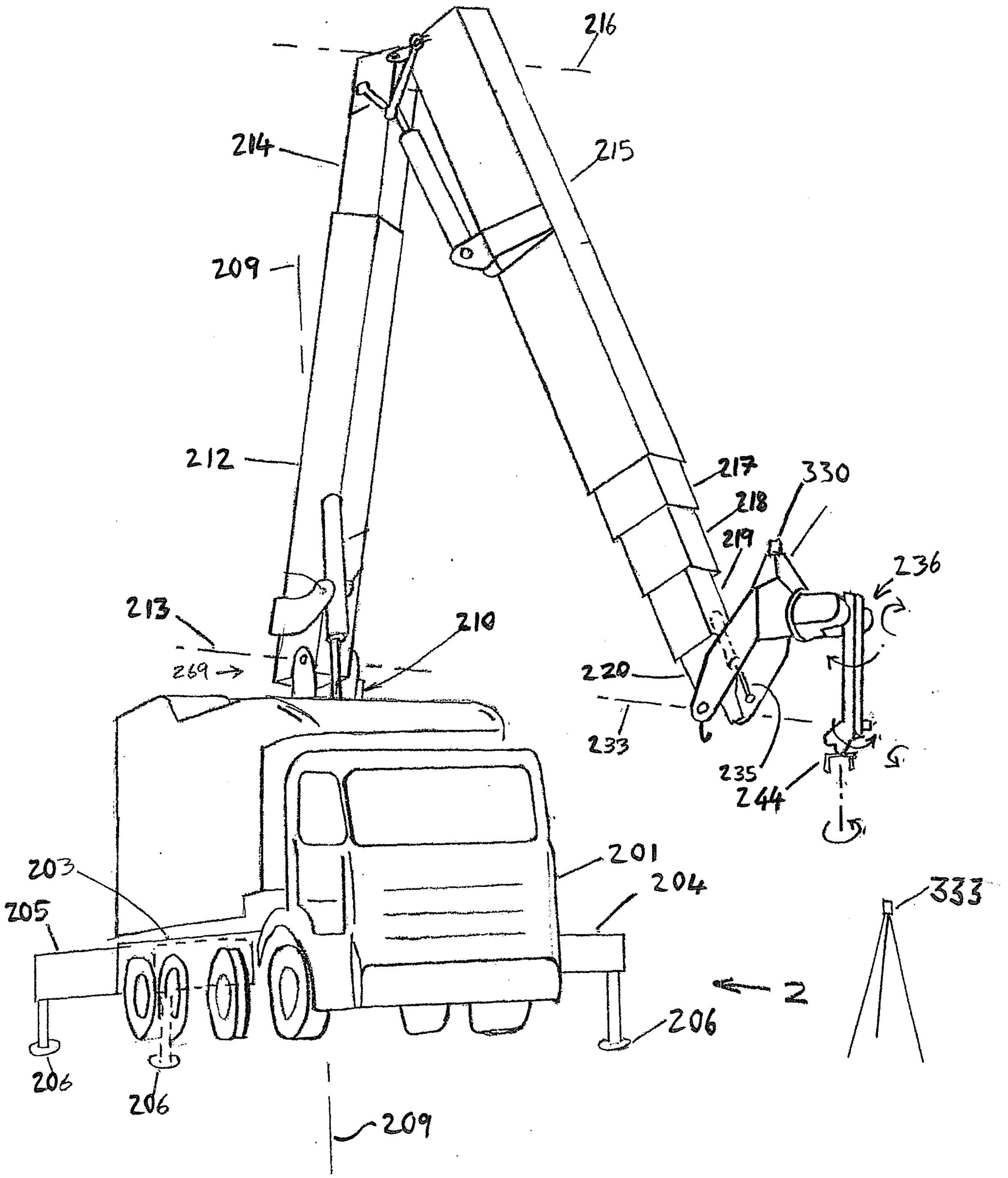
216
214
215
209
212
217
330
218
219
236
213
210
269
220
233
235
244
201
203
204
205
333
2
206
206
206
209

FIGURE 23

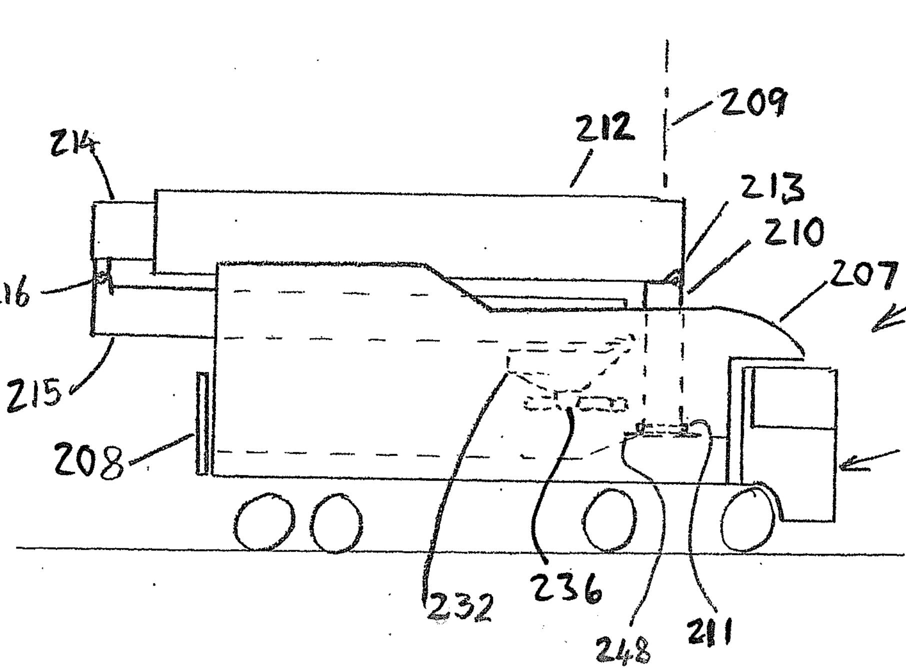

FIGURE 24

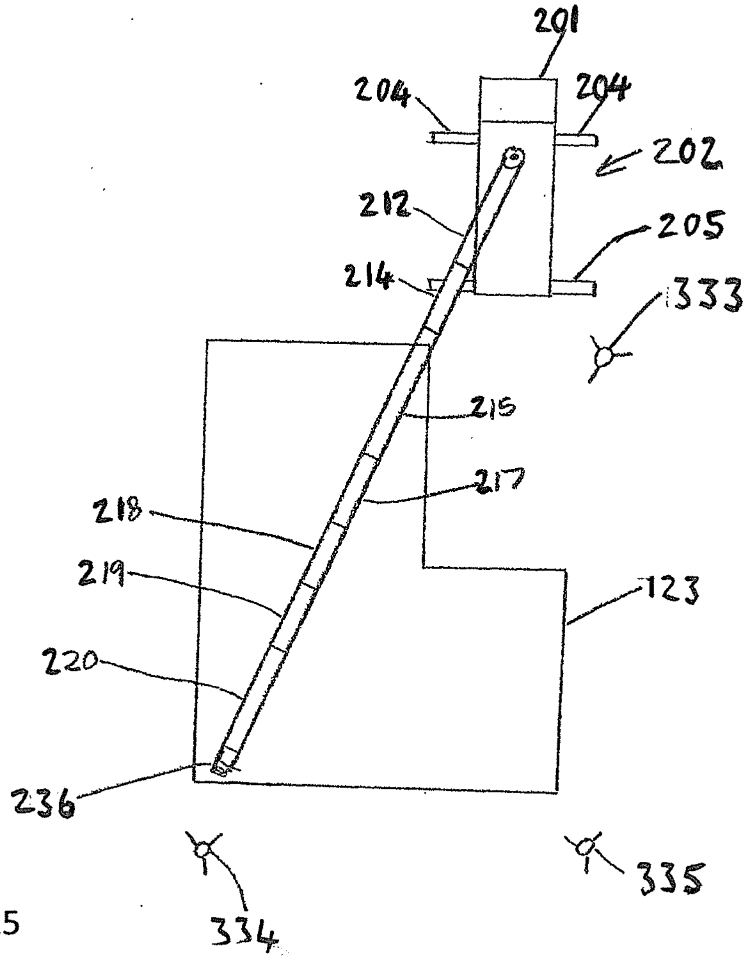

FIGURE 25

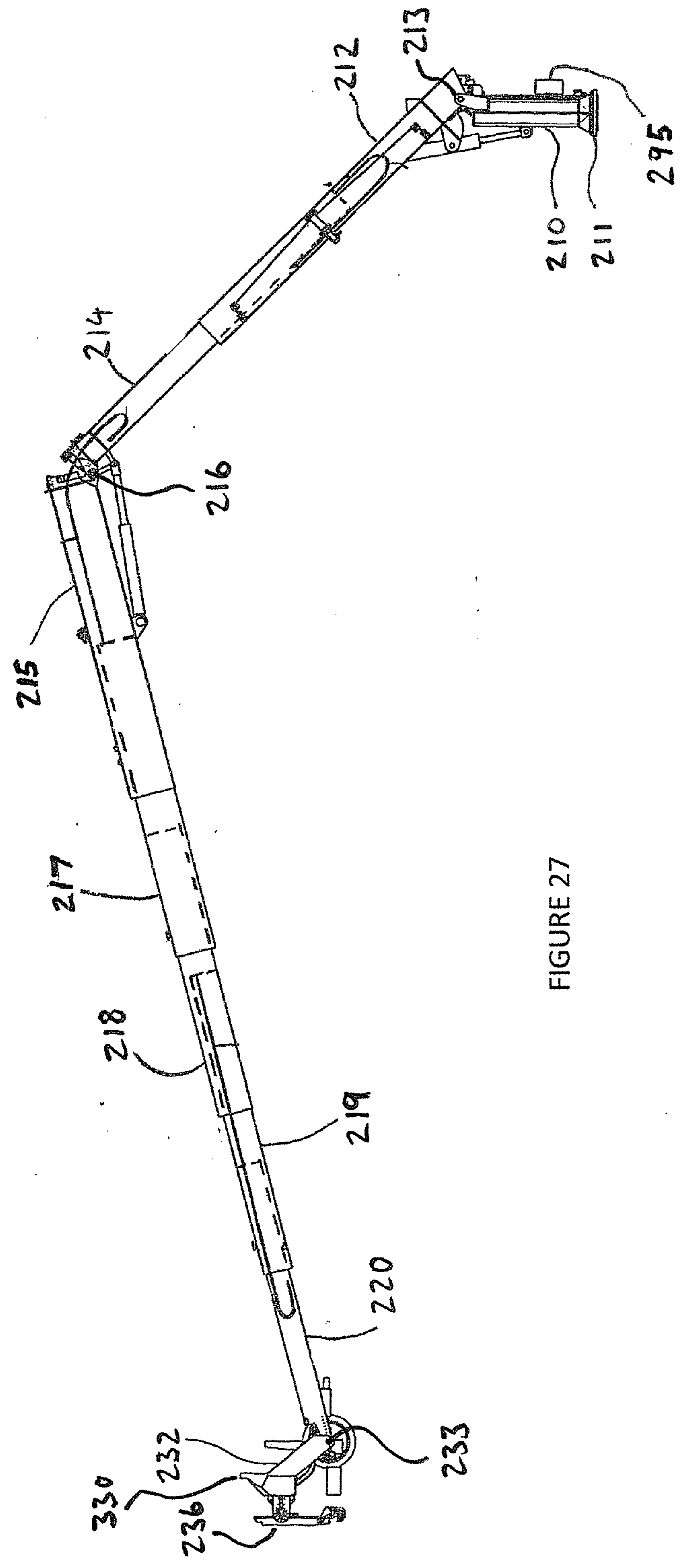

FIGURE 27

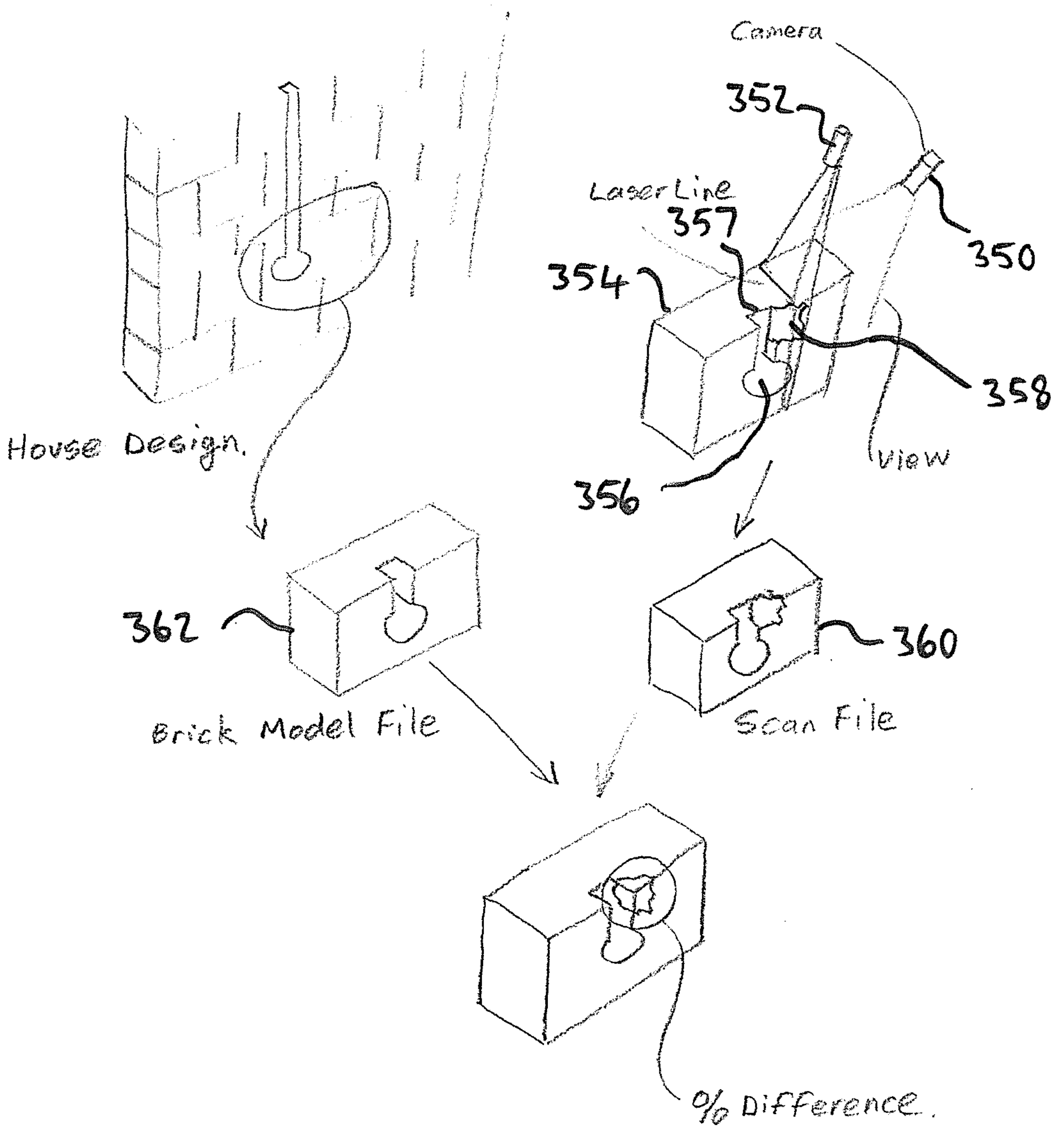
Camera
352
Laser Line
357
350
354
358
House Design.
View
356
362
360
Brick Model File
Scan File
% Difference.

FIGURE 28

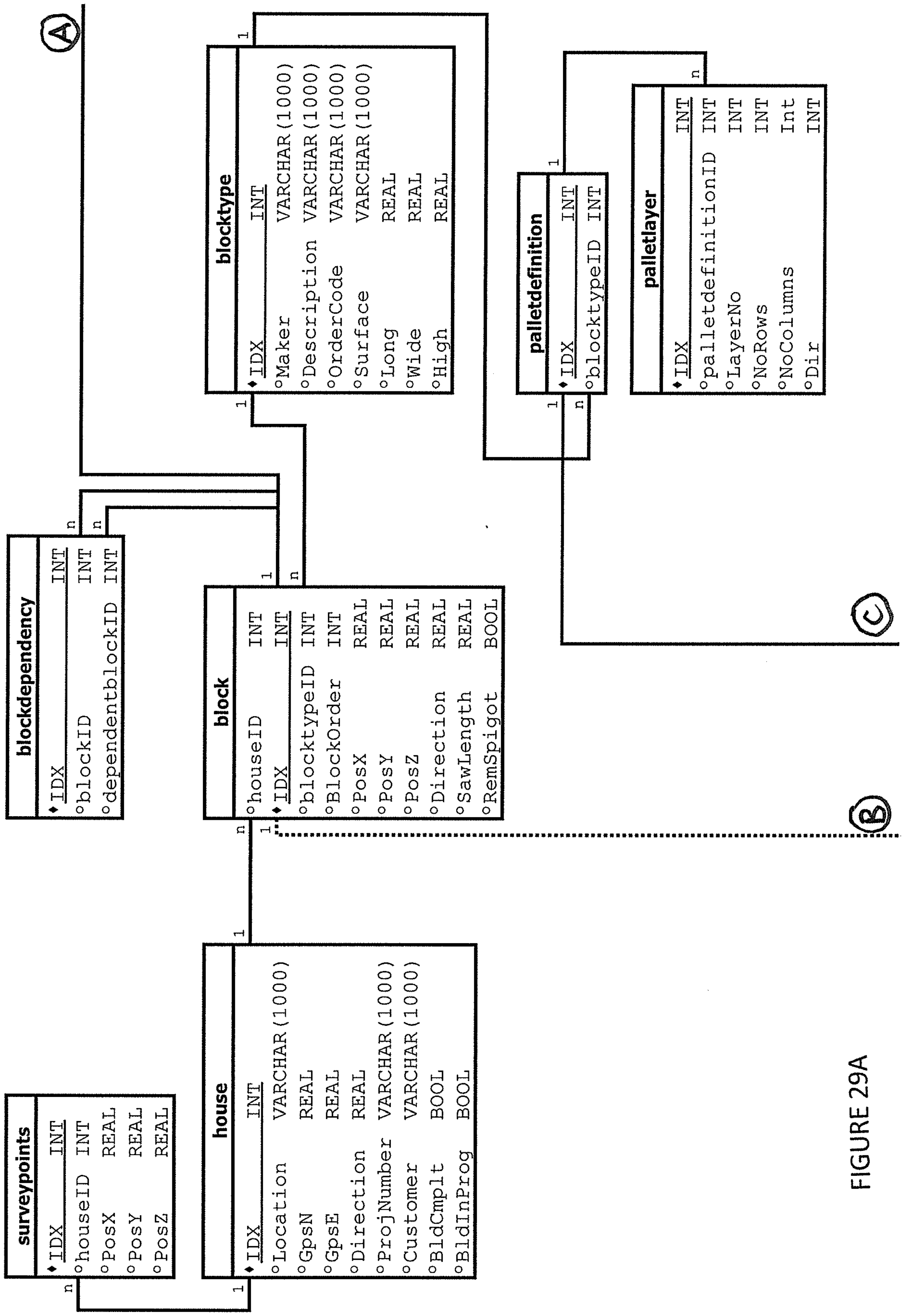
A
blocktype
IDX INT
Maker VARCHAR(1000)
Description VARCHAR(1000)
OrderCode VARCHAR(1000)
Surface VARCHAR(1000)
Long REAL
Wide REAL
High REAL
palletdefinition
IDX INT
blocktypeID INT
palletlayer
IDX INT
palletdefinitionID INT
LayerNo INT
NoRows INT
NoColumns Int
Dir INT
blockdependency
IDX INT
blockID INT
dependentblockID INT
block
houseID INT
IDX INT
blocktypeID INT
BlockOrder INT
PosX REAL
PosY REAL
PosZ REAL
Direction REAL
SawLength REAL
RemSpigot BOOL
surveypoints
IDX INT
houseID INT
PosX REAL
PosY REAL
PosZ REAL
house
IDX INT
Location VARCHAR(1000)
GpsN REAL
GpsE REAL
Direction REAL
ProjNumber VARCHAR(1000)
Customer VARCHAR(1000)
BldCmplt BOOL
BldInProg BOOL
B
C

FIGURE 29A

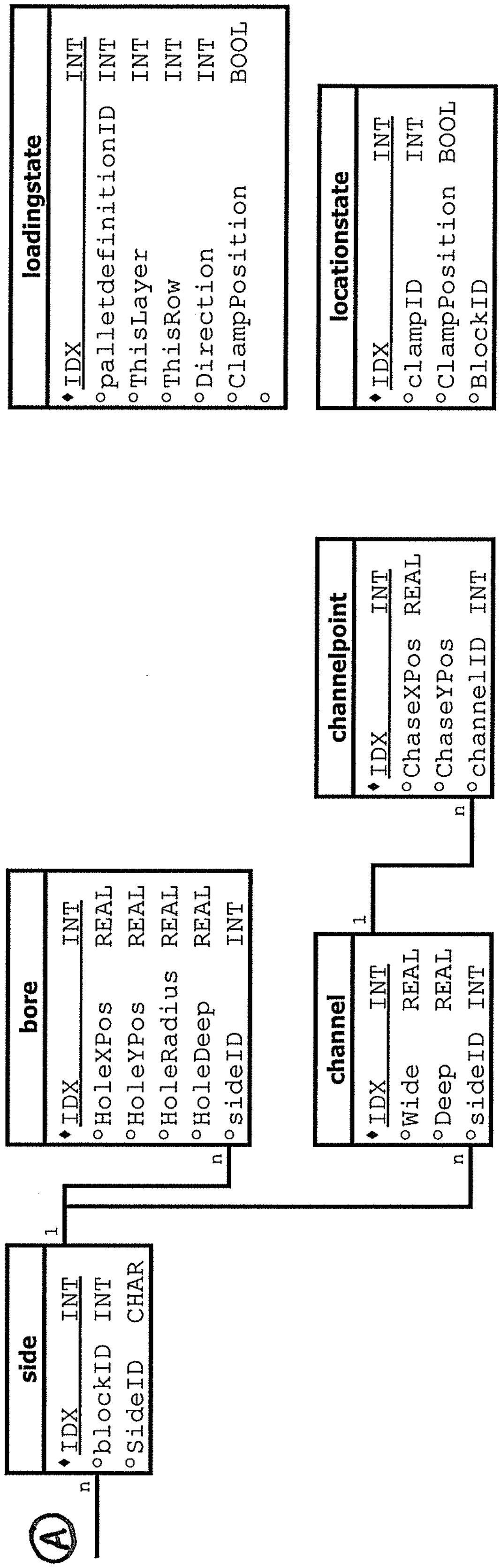
loadingstate
IDX INT
palletdefinitionID INT
ThisLayer INT
ThisRow INT
Direction INT
ClampPosition BOOL
locationstate
IDX INT
clampID INT
ClampPosition BOOL
BlockID
channelpoint
IDX INT
ChaseXPos REAL
ChaseYPos
channelID INT
bore
IDX INT
HoleXPos REAL
HoleYPos REAL
HoleRadius REAL
HoleDeep REAL
sideID INT
channel
IDX INT
Wide REAL
Deep REAL
sideID INT
side
IDX INT
blockID INT
SideID CHAR
A
n
1
n
n
1
n

FIGURE 29B

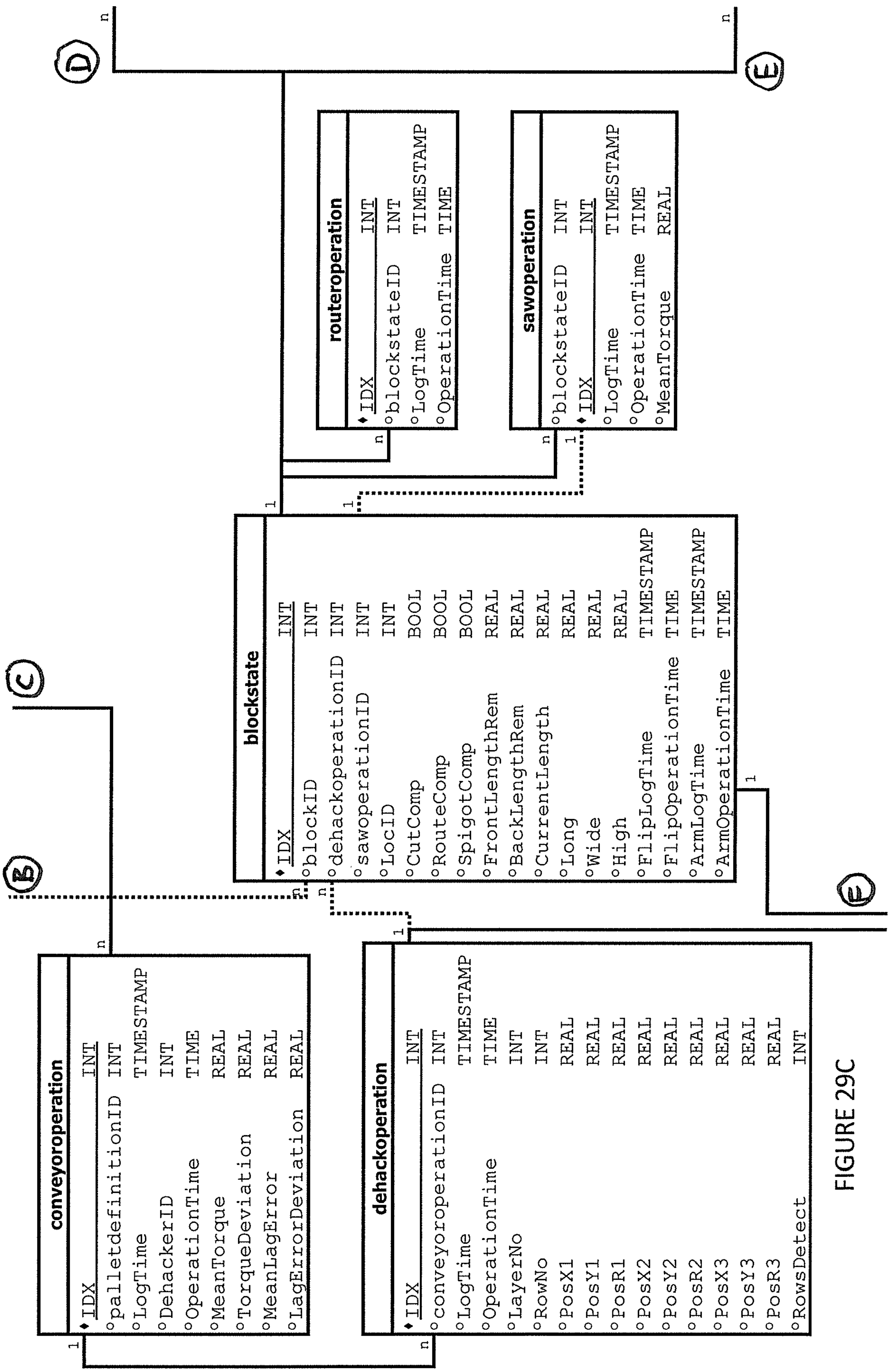
D
E
C
B
F
routeroperation
IDX INT
blockstateID INT
LogTime TIMESTAMP
OperationTime TIME
sawoperation
blockstateID INT
IDX INT
LogTime TIMESTAMP
OperationTime TIME
MeanTorque REAL
blockstate
IDX INT
blockID INT
dehackoperationID INT
sawoperationID INT
LocID INT
CutComp BOOL
RouteComp BOOL
SpigotComp BOOL
FrontLengthRem REAL
BackLengthRem REAL
CurrentLength REAL
Long REAL
Wide REAL
High REAL
FlipLogTime TIMESTAMP
FlipOperationTime TIME
ArmLogTime TIMESTAMP
ArmOperationTime TIME
conveyoroperation
IDX INT
palletdefinitionID INT
LogTime TIMESTAMP
DehackerID INT
OperationTime TIME
MeanTorque REAL
TorqueDeviation REAL
MeanLagError REAL
LagErrorDeviation REAL
dehackoperation
IDX INT
conveyoroperationID INT
LogTime TIMESTAMP
OperationTime TIME
LayerNo INT
RowNo INT
PosX1 REAL
PosY1 REAL
PosR1 REAL
PosX2 REAL
PosY2 REAL
PosR2 REAL
PosX3 REAL
PosY3 REAL
PosR3 REAL
RowsDetect INT
n
1

FIGURE 29C

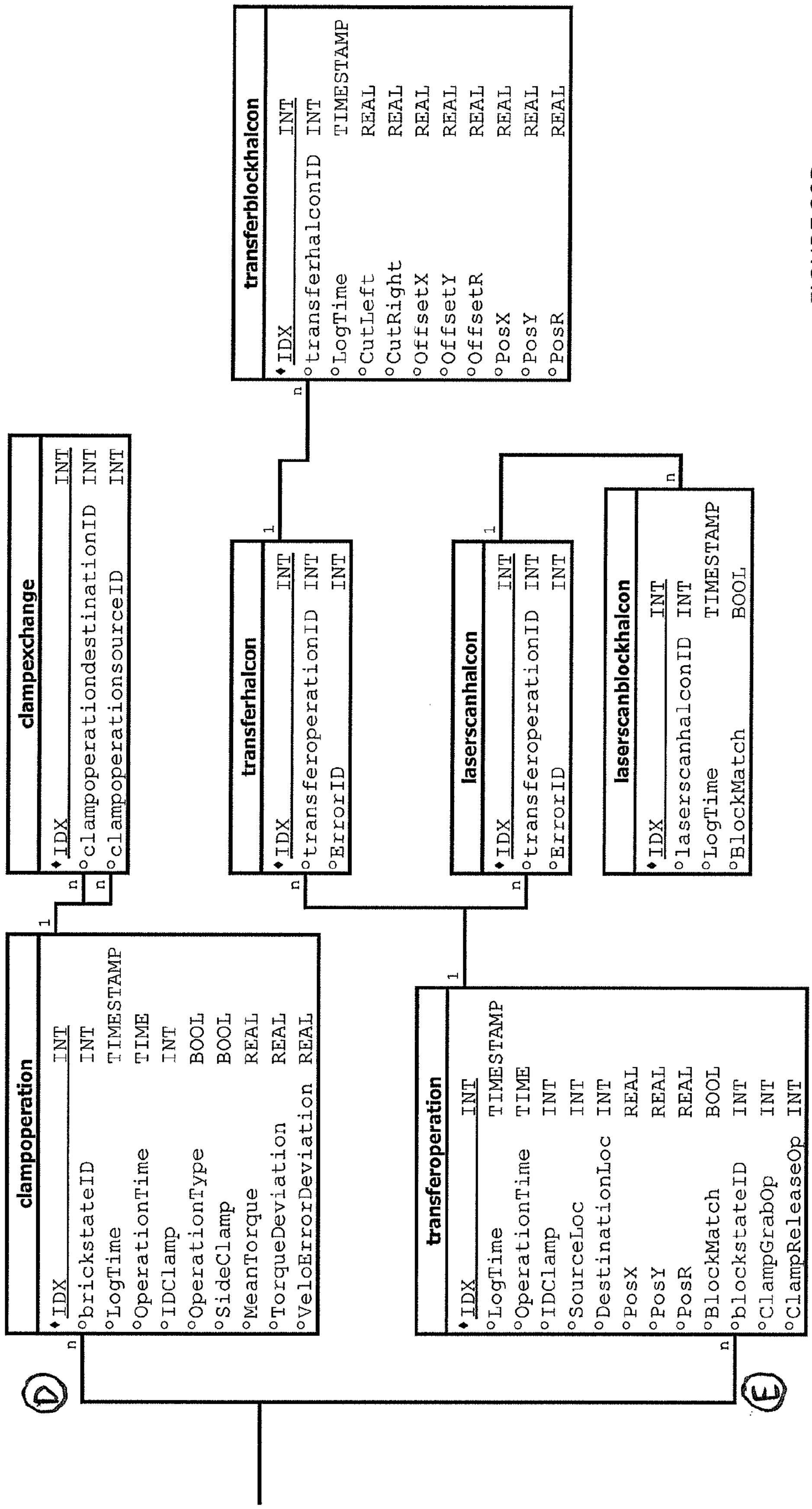
clampoperation
IDX INT
brickstateID INT
LogTime TIMESTAMP
OperationTime TIME
IDClamp INT
OperationType BOOL
SideClamp BOOL
MeanTorque REAL
TorqueDeviation REAL
VeloErrorDeviation REAL
clampexchange
IDX INT
clampoperationdestinationID INT
clampoperationsourceID INT
transferoperation
IDX INT
LogTime TIMESTAMP
OperationTime TIME
IDClamp INT
SourceLoc INT
DestinationLoc INT
PosX REAL
PosY REAL
PosR REAL
BlockMatch BOOL
blockstateID INT
ClampGrabOp INT
ClampReleaseOp INT
transferhalcon
IDX INT
transferoperationID INT
ErrorID INT
laserscanhalcon
IDX INT
transferoperationID INT
ErrorID INT
laserscanblockhalcon
IDX INT
laserscanhalconID INT
LogTime TIMESTAMP
BlockMatch BOOL
transferblockhalcon
IDX INT
transferhalconID INT
LogTime TIMESTAMP
CutLeft REAL
CutRight REAL
OffsetX REAL
OffsetY REAL
OffsetR REAL
PosX REAL
PosY REAL
PosR REAL
D
E
1
n

FIGURE 29D

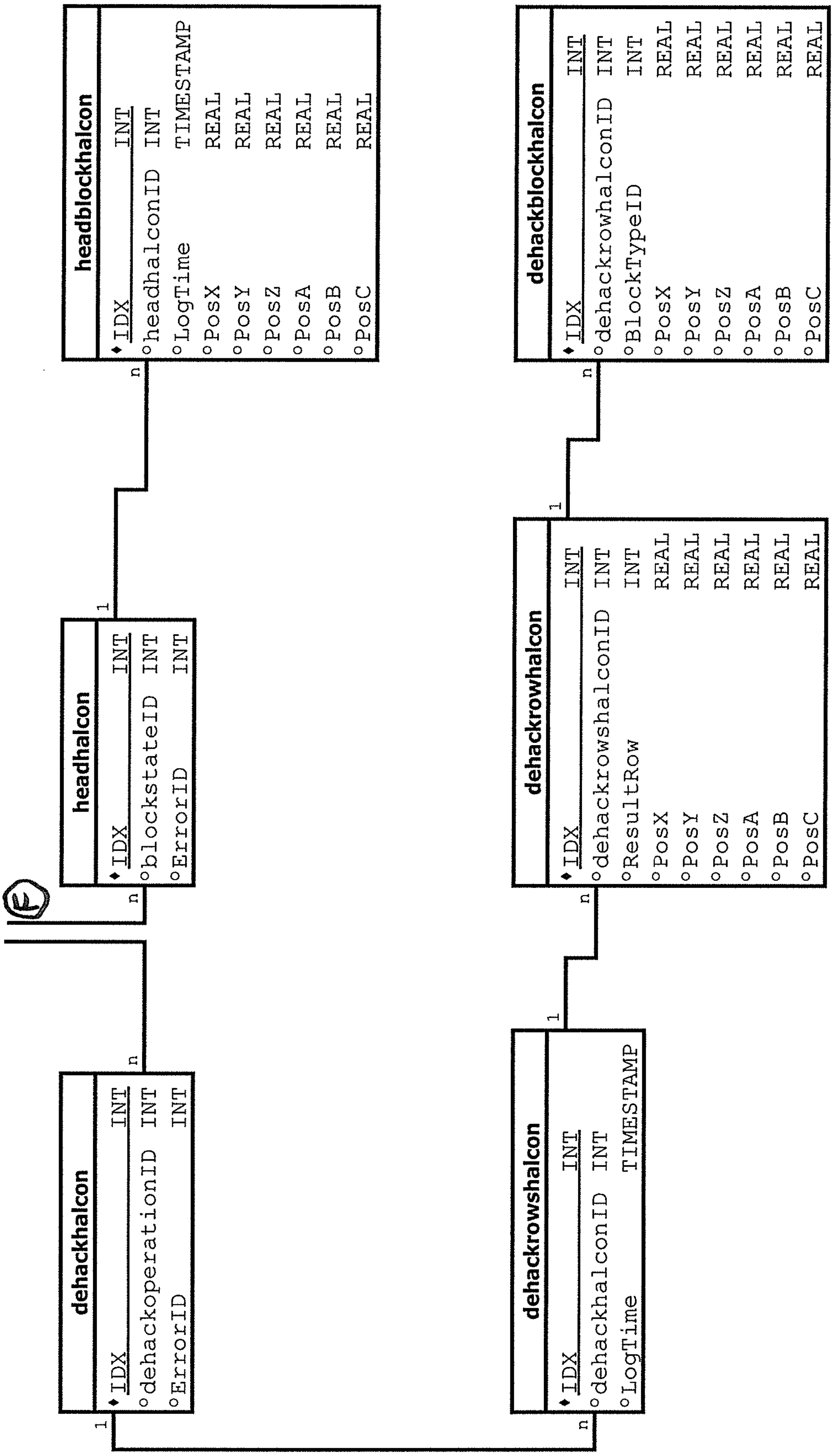
headblockhalcon
IDX INT
headhalconID INT
LogTime TIMESTAMP
PosX REAL
PosY REAL
PosZ REAL
PosA REAL
PosB REAL
PosC REAL
headhalcon
IDX INT
blockstateID INT
ErrorID INT
dehackhalcon
IDX INT
dehackoperationID INT
ErrorID INT
dehackblockhalcon
IDX INT
dehackrowhalconID INT
BlockTypeID INT
PosX REAL
PosY REAL
PosZ REAL
PosA REAL
PosB REAL
PosC REAL
dehackrowhalcon
IDX INT
dehackrowshalconID INT
ResultRow INT
PosX REAL
PosY REAL
PosZ REAL
PosA REAL
PosB REAL
PosC REAL
dehackrowshalcon
IDX INT
dehackhalconID INT
LogTime TIMESTAMP
F
1
n

FIGURE 29E

drive	
♦IDX	INT
°DriveID	INT
°LogTime	TIMESTAMP
°TotalDist	BIGINT
°ExecutePosition	BOOL
°PositionSP	REAL
°Position	REAL
°PositionLag	REAL
°Velocity	REAL
°Acceleration	REAL
°Torque	REAL

drivereference	
♦IDX	INT
°DriveID	INT
°LogTime	TIMESTAMP
°TotalDist	BIGINT
°ExecutePosition	BOOL
°PositionSP	REAL
°Position	REAL
°PositionLag	REAL
°Velocity	REAL
°Acceleration	REAL
°Torque	REAL
°ExecutePositionRef	BOOL
°PositionRefSP	REAL
°PositionRef	REAL
°VelocityRef	REAL
°AccelerationRef	REAL

drivereference	
♦IDX	INT
°DriveID	INT
°LogTime	TIMESTAMP
°TotalDist	BIGINT
°ExecutePosition	BOOL
°PositionSP	REAL
°Position	REAL
°PositionLag	REAL
°Velocity	REAL
°Acceleration	REAL
°Torque	REAL
°ExecutePositionRef	BOOL
°PositionRefSP	REAL
°PositionRef	REAL
°VelocityRef	REAL
°AccelerationRef	REAL
°ClampTotalDistance	BIGINT
°OperationState	INT

FIGURE 29F

INTERNATIONAL SEARCH REPORT

International application No.
PCT/AU2017/050738

A. CLASSIFICATION OF SUBJECT MATTER

G06F 17/50 (2006.01) G06F 17/30 (2006.01) G05B 19/4097 (2006.01) E04G 21/22 (2006.01)

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPOQUE PATENW: IPC/CPC - G06F17/50, G06F2217/00, G06Q10/04, G05B19/4097, G05B2219/45086, G06F9/00, E04B2/02, E04C1/00, E04G21/22; Keywords - brick, construction, building, control, system, software, size, dimensions, placement, plan, walls, masonry, fabrication, database, footings, configuration, data, laying, 3-D, CAD, cut, machining & similar terms.
Google Patents: Keywords - G06F17/50, brick, plan, wall, placement, arrangement, location, position, CAD, plan, CAD-CAM, brickwork; control, system, software, program, app, application, code, instruction, brick, masonary; method, building, brick, placement, top-plan, walls, dimensions, 3-D.
Google Scholar: Keywords - CAD, building, plan, brick, wall, placement, CAD-CAM, brickwork; robotic, bricklaying, control system.
Applicant & Inventor Name Search: performed in Espacenet, Google Patents & IP Australia internal databases.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
	Documents are listed in the continuation of Box C	

[X] Further documents are listed in the continuation of Box C [X] See patent family annex

*	Special categories of cited documents:		
"A"	document defining the general state of the art which is not considered to be of particular relevance	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"E"	earlier application or patent but published on or after the international filing date	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O"	document referring to an oral disclosure, use, exhibition or other means	"&"	document member of the same patent family
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search	Date of mailing of the international search report
17 October 2017	17 October 2017

Name and mailing address of the ISA/AU

AUSTRALIAN PATENT OFFICE
PO BOX 200, WODEN ACT 2606, AUSTRALIA
Email address: pct@ipaustralia.gov.au

Authorised officer

Ravi McCosker
AUSTRALIAN PATENT OFFICE
(ISO 9001 Quality Certified Service)
Telephone No. +612 6283 2933

INTERNATIONAL SEARCH REPORT		International application No.
C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		**PCT/AU2017/050738**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
	WO 2011/077006 A2 (IBRIQ OY) 30 June 2011	
Y	Entire document, especially: Abstract; Fig. 1-4; Page 3:lines 6-30, Page 6:lines 1-24, Page 9:lines 9-21.	1-16
A		19-31
	US 2005/0131619 A1 (RAPPAPORT et al.) 16 June 2005	
Y	Entire document, especially: Abstract, Para. 3, 7, 8, 23, 24, 28, 40, 42, 64, 72, 79.	1-16
A		19-31
	US 2009/0038258 A1 (PIVAC et al.) 12 February 2009	
X	Entire document, especially: Abstract; Fig. 6; Para. 8-12; 16, 40, 41, 49, 50, 54, 55, 59, 63-65, 72.	19-26, 28-31
Y	Entire document, especially: Abstract; Para. 16, 40, 49, 72.	1-16
A		17, 18, 27
A	LATTEUR et al., Drone-Based Additive Manufacturing of Architectural Structures, August 2015 [retrieved from internet on 10 October 2017] <URL: https://www.researchgate.net/publication/316093173_Drone-Based_Additive_Manufacturing_of_Architectural_Structures >	1-18
A	US 2015/0082740 A1 (CONSTRUCTION ROBOTICS LLC) 26 March 2015	19-25, 29-31
A	US 2014/0366481 A1 (RICHARD MARK BENSON) 18 December 2014	19-25, 26-31
A	WO 2013/088154 A1 (INTELLIGENT BUILDING PROCESSES LTD) 20 June 2013	26-28
A	EP 2112291 A1 (FOLIEROL VOF) 28 October 2009	26-28

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:

   because they relate to subject matter not required to be searched by this Authority, namely:

   the subject matter listed in Rule 39 on which, under Article 17(2)(a)(i), an international search is not required to be carried out, including

2. ☐ Claims Nos.:

   because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos:

   because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a)

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

See Supplemental Box for Details

1. ☒ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☒ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.

Supplemental Box

Continuation of: Box III
This International Application does not comply with the requirements of unity of invention because it does not relate to one invention or to a group of inventions so linked as to form a single general inventive concept.

This Authority has found that there are different inventions based on the following features that separate the claims into distinct groups:

- Group 1: Claims 1-18 are directed to CAD software for designing a brick construction. The feature of the building plan data is specific to this group of claims.

- Group 2: Claims 19-25, 29-31 are directed to control software for control of a brick laying machine. The feature of the brick laying machine is specific to this group of claims.

- Group 3: Claims 26-28 are directed to a method of building a structure. The feature of the footings is specific to this group of claims.

PCT Rule 13.2, first sentence, states that unity of invention is only fulfilled when there is a technical relationship among the claimed inventions involving one or more of the same or corresponding special technical features. PCT Rule 13.2, second sentence, defines a special technical feature as a feature which makes a contribution over the prior art.

When there is no special technical feature common to all the claimed inventions there is no unity of invention.

In the above groups of claims, the identified features may have the potential to make a contribution over the prior art but are not common to all the claimed inventions and therefore cannot provide the required technical relationship. The only feature common to all of the claimed inventions and which provides a technical relationship among them is a *'brick placement database'*.

However this feature does not make a contribution over the prior art because it is disclosed in:

D3: US 2009/0038258 A1 (PIVAC et al.) 12 February 2009 [see D3: Para. 40, 72, 74].

Therefore in the light of this document this common feature cannot be a special technical feature. Therefore there is no special technical feature common to all the claimed inventions and the requirements for unity of invention are consequently not satisfied *a posteriori*.

Furthermore, in relation to:

- Group 2: Claims 19-25, 29-31 are directed to control software for control of a brick laying machine. The feature of the brick laying machine is specific to this group of claims.

- Group 3: Claims 26-28 are directed to a method of building a structure. The feature of the footings is specific to this group of claims.

The only feature common to all of the claimed inventions and which provides a technical relationship among them is a *'brick placement database'* & *'apply adhesive and locate each said brick according to data stored in said brick placement database'*.

However these features do not make a contribution over the prior art because they are disclosed in:

D3: US 2009/0038258 A1 (PIVAC et al.) 12 February 2009 [see D3: Para. 40, 42, 55, 72, 74].

Therefore in the light of this document these common features cannot be special technical features. Therefore there is no special technical feature common to all the claimed inventions and the requirements for unity of invention are consequently not satisfied *a posteriori*.

Furthermore, in relation to:

- Group 2: Claims 20-25, 29-31 are directed to control software for control of a brick laying machine. The feature of the brick laying machine is specific to this group of claims.

- Group 3: Claims 26-28 are directed to a method of building a structure. The feature of the footings is specific to this group of claims.

The only features common to all of the claimed inventions and which provides a technical relationship among them is *'trim data in the form of a trim value or trim value array'*, *'brick placement database'* & *'apply adhesive and locate each said brick according to data stored in said brick placement database'*.

However these features do not make a contribution over the prior art because they are disclosed in:

Supplemental Box

D3: US 2009/0038258 A1 (PIVAC et al.) 12 February 2009 [see D3: Para. 40, 42, 55, 72, 74 – cutting is considered trimming; trim data in the form of a trim value is considered inherent].

Therefore in the light of this document these common features cannot be special technical features. Therefore there is no special technical feature common to all the claimed inventions and the requirements for unity of invention are consequently not satisfied *a posteriori*.

Furthermore, in relation to:

- Group 2: Claims 20, 22-25, 29-31 are directed to control software for control of a brick laying machine. The feature of the brick laying machine is specific to this group of claims.

- Group 3: Claims 26-28 are directed to a method of building a structure. The feature of the footings is specific to this group of claims.

The only features common to all of the claimed inventions and which provides a technical relationship among them is a '*trim data in the form of a trim value or trim value array*' & '*an amount to be machined from a horizontal face of each brick so that the top of a course of bricks is level when laid*', '*brick placement database*' & '*apply adhesive and locate each said brick according to data stored in said brick placement database*'.

However these features do not make a contribution over the prior art because they are disclosed in:

D3: US 2009/0038258 A1 (PIVAC et al.) 12 February 2009 [see D3: Para. 40, 42, 55, 72, 74 – cutting is considered trimming; trim data in the form of a trim value is considered inherent];

and the feature '*an amount to be machined from a horizontal face of each brick so that the top of a course of bricks is level when laid*' is considered generic in this particular art.

Therefore in the light of this document these common features cannot be special technical features. Therefore there is no special technical feature common to all the claimed inventions and the requirements for unity of invention are consequently not satisfied *a posteriori*.

INTERNATIONAL SEARCH REPORT Information on patent family members	International application No. **PCT/AU2017/050738**

This Annex lists known patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent Document/s Cited in Search Report		**Patent Family Member/s**	
Publication Number	**Publication Date**	**Publication Number**	**Publication Date**
WO 2011/077006 A2	30 June 2011	WO 2011077006 A2	30 Jun 2011
		EP 2516762 A2	31 Oct 2012
		FI 20096381 A	23 Jun 2011
		FI 124179 B	15 Apr 2014
US 2005/0131619 A1	16 June 2005	US 2005131619 A1	16 Jun 2005
		AU 5004300 A	18 Dec 2000
		US 6721769 B1	13 Apr 2004
		US 6850946 B1	01 Feb 2005
		US 2004177085 A1	09 Sep 2004
		US 7711687 B2	04 May 2010
		US 2004162840 A1	19 Aug 2004
		US 2004181374 A1	16 Sep 2004
		US 2004186847 A1	23 Sep 2004
		WO 0073874 A2	07 Dec 2000
US 2009/0038258 A1	12 February 2009	US 2009038258 A1	12 Feb 2009
		US 8166727 B2	01 May 2012
		AU 2007203730 A1	12 Jul 2007
		AU 2007203730 B2	27 Sep 2012
		CA 2633733 A1	12 Jul 2007
		CN 101360873 A	04 Feb 2009
		CN 101360873 B	04 Jul 2012
		EP 1977058 A1	08 Oct 2008
		EP 1977058 B1	16 Jul 2014
		JP 2009521630 A	04 Jun 2009
		WO 2007076581 A1	12 Jul 2007
US 2015/0082740 A1	26 March 2015	US 2015082740 A1	26 Mar 2015
		US 2012053726 A1	01 Mar 2012
		US 8965571 B2	24 Feb 2015
		US 2017254102 A1	07 Sep 2017
		WO 2012021732 A1	16 Feb 2012
US 2014/0366481 A1	18 December 2014	US 2014366481 A1	18 Dec 2014
		US 8825208 B1	02 Sep 2014

Due to data integration issues this family listing may not include 10 digit Australian applications filed since May 2001.

INTERNATIONAL SEARCH REPORT Information on patent family members	International application No. **PCT/AU2017/050738**

This Annex lists known patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent Document/s Cited in Search Report		**Patent Family Member/s**	
Publication Number	**Publication Date**	**Publication Number**	**Publication Date**
WO 2013/088154 A1	20 June 2013	WO 2013088154 A1	20 Jun 2013
		GB 2497537 A	19 Jun 2013
		GB 2497537 B	09 Jul 2014
EP 2112291 A1	28 October 2009	EP 2112291 A1	28 Oct 2009

End of Annex

Due to data integration issues this family listing may not include 10 digit Australian applications filed since May 2001.