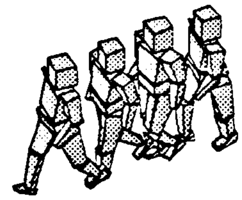
ANALYZE



An Overview

What is ANALYZE ?

ANALYZE is an analysis package for processing motion, force plate and EMG data from a motion measurement system which can generate a number of quantities that are used for diagnosis and research of human locomotion and locomotion disorders as well as other activities such as chair raising or upper extremity motion. In fact, at present *ANALYZE* will generate over 100 different quantities. The input data required to use *ANALYZE* are 3 dimensional marker trajectories, force plate ground reaction load data and/or muscle electromyography data. The previous senence uses 'and/or' because *ANALYZE* is capable of performing either motion, ground reaction, or EMG analyses alone or in combination with one another. This enables marker data to be crunched to get purely kinematic information, such as joint angles, velocities, and accelerations without having to know force plate data (obviously that data isn't needed for such an analysis). It also enables ground reaction force data to be crunched to get such purely load related information as center of pressure and ground impulse. But, since all this information is required to find joint moments and powers (in reality this is only true for the double support phase of gait- more on that later), *ANALYZE* has been set up in a flexible fashion such that the tools which need to be available to get these quantities are also available for use in studies which have little or nothing to do with gait. For example, the force plate section of *ANALYZE* can be used to produce center of pressure information on a person's balance as they stand for a period of time on the force plate(s). The EMG section can be used to perform signal processing tasks such as digital filtering, integration, and rectification. The results can be displayed in a multitude of ways in *TELIO*, the graphics postprocessing program that is a companion to *ANALYZE*. *TELIO* is discussed in detail in its own documentation. In addition to full body motion, *ANALYZE* has been configured for studying upper extremity motion in which only one arm and the trunk and head are present. Another example is that rigid body arrays can be tested using the motion section of *ANALYZE*. In the end, the result is that one package can be used to study a large number of motion, EMG, and ground load protocols without having to create special programs for every little variation that one comes up with, as has been done in the past.

In addition, a consistent and extendible file format has been developed to store the analysis results in a machine independent format. In this way, the results will be usable on any future machines which a lab may purchase or decide to use for other reasons. This also facilitates the sharing of data between researchers. The format is extendible in that if future researchers come up with other analyses to add to *ANALYZE* (which is relatively easy to do and will most definitely happen) the resulting files will still be compatible with previously written program code used to read *ANALYZE* files. Up till now every time another analysis method was developed another file format was created and more programs needed to be written to use the files. The result has been a proliferation of files, each of a unique, idiosyncratic type, which a programmer needed to wrestle with for a while. Well, no more!

Future additions to *ANALYZE* have been planned for by the very way in which it has been designed internally. Basically, the analysis code is based upon the principles of the mechanics of three dimensional rigid bodies. These analyses do not contain any simplifications which prevent their use in future studies (actually there is one dealing with the inertia matrix of a body segment. More on that in a later section.), as is the case with software which is oriented toward sagittal plane motions only and hence contains two dimensional motion assumptions. To do this, *ANALYZE* has been structured to perform its analysis upon only 3D rigid body segments (e.g., the foot, calf, and thigh) which are connected by joints which do not have any constrained degrees of freedom. In otherwards, the segments are free to move relative to one another (3 translations and 3 rotations- the definition of these is discussed later in the section about coordinate systems). In fact, whether the whole body, only the upper body, or only the lower body is present becomes immaterial to the analysis routines as long as the segment connections (i.e., joints) have been defined properly.

For this type of flexibility to take place, the first thing that *ANALYZE* does is use the marker data to calculate the position and orientation of the body segments in space in each frame of data. In addition, the joint position relative to these segments must be defined (in the segment's local coordinate system) and the segment mass and inertia calculated from anthropometric data. (For mathematical purists out there, yes the joint positions are assumed based upon the anthropometric data, BUT the kinematic screws of the relative motion between segments can be calculated in the subsequent analyses and could be used to find the "joint axis". This may be a future enhancement, but don' t bug me about it.) This is the only part of *ANALYZE* which is marker set specific and thus these routines are the only ones that need to be modified if a different marker set is used. In fact, *ANALYZE* has been set up such that it can support a number of marker sets by using option switches in its commands. (See the later section about how to modify *ANALYZE* for your own marker set.) A better solution would be to have the marker set to segment calculations in a parameter file which would be read and interpreted by *ANALYZE*. The calculations would essentially be mini programs in some compact language. This is very desirable, but is path fraught with pitfalls, traps, and headaches (e.g., what should *ANALYZE* do if the person who wrote the marker to segment calculation did it so wrong that divide by 0's occur ?) and deadlines needed to be met. Thus, it is not part of *ANALYZE* at present but it is definitely a future enhancement (once again, don' t bug me about when it will be done- I' m not a computer science major so I need to learn how write a language interpreter, etc...). Once the marker to segment calculations have been performed the only other thing that *ANALYZE* needs to know is how many segments there are, which ones they are, and how they are connected together.

What does ANALYZE recognize ?

At present, *ANALYZE* is set up to recognize up to 15 unique body segments and 15 unique joints. It is 'up to' because, as pointed out previously, the program has been set up in flexible manner so that not all segments and joints need be included in the analyses. However, ***care must be taken at this point.*** If you choose to analyze less than the full body, then the analysis must be set up correctly. For example, the joint angles of the shoulder can't be calculated from the position and orientation of the thigh and calf. *ANALYZE* is not set up to be used by just anyone. The user must understand the principles of mechanics to use and/or modify the program properly.

The body segments which *ANALYZE* recognizes and their abbreviations are:

Right Foot	rtft	Left Foot	lfft
Right Calf	rtcf	Left Calf	lfcf
Right Thigh	rtth	Left Thigh	lfth
Pelvis	pelv		
Trunk	trnk		
Right Hand	rthd	Left Hand	lfhd
Right Forearm	rtla	Left Forearm	lfla
Right Upperarm	rtua	Left Upperarm	lfua
Head	head		

The joints which *ANALYZE* recognizes are:

Right Ankle	rtak	Left Ankle	lfak
Right Knee	rtkn	Left Knee	lfkn
Right Hip	rthp	Left Hip	lfhp
Pelvis-LAB GCS	pvlb		
Pelvis-Trunk	pvtk		
Right Wrist	rtwr	Left Wrist	lfwr
Right Elbow	rtel	Left Elbow	lfel
Right Shoulder	rtsh	Left Shoulder	lfsh
Neck	neck		

Note that the pvlb joint is only used to compute the absolute pelvis angle and thus can be thought of as an imaginary joint since there is no such anatomic joint. This joint is the only one in which the joint angles are calculated relative the global coordinate system (GCS). In all other joints, the angles are defined as those between the segment distal to the joint relative to the proximal segment.

Other joints could be added later on, but the arrays within the program have been set up with these limits (15 segments, 15 joints) in mind. The file format is set up in such a way that other segments and joints could be added and the files could still be read by subsequent programs, such as *TELIO*, without having to rewrite the file reading code. However, the program which reads such a file must have provision for larger arrays and a way to use this extra data. *TELIO* is set up to handle the 15 segments and 15 joints defined here at a maximum. (As an aside, *TELIO* can read in up to 6 different ANZ result files and plot any combination of their contents in graphs or display the contents in 3D. So, even though there is a segment and joint limit, it can still do alot.)

What can ANALYZE calculate ?

The quantities that *ANALYZE* calculates can be grouped into several categories: marker based information, gait cycle time-distance information, segment based information, joint based information,force plate based information, and EMG information. Each of these is describedin the following sections.

Marker based information

Marker position: Marker position data read in from a file can be checked for gaps and interpolated/extrapolated using a selectable order polynomial. Marker data can then be smoothed if desired with any of three methods. A generalized cross validated smoothing polynomial (GCV) (Woltring, 1986) which automatically adapts to the data, a dynamic programming filter (DPF) (Dohrman, 1988) which is also adaptive, and digital filtering with selectable high, low and notch filter frequencies. NOTE: *ANALYZE* has the capability to interpolate the marker data onto any specified interval with any number samples created on that interval. This is convienient for making gait data appear at regular intervals of the gait cycle, for instance every 2% of the cycle.

Marker velocity and acceleration: The first and second derivatives of the marker trajectories can be calculated using either the GCV, DPF, or finite difference methods.

Marker diagnostics: Used to test the accuracy, resolution, and repeatability of the motion measurement system. This is essential in determining limits of confidence in the analysis results. Two measures are provided: 1) The intermarker distances between markers on the same segment in each frame of motion data and 2) the position of the segment markers relative to the segment local coordinate system. These show how the markers are moving relative to one another within a segment. Ideally there should be no movement if the markers were rigidly attached to a rigid body but in reality there is motion due to such factors as skin and muscle motion and vibration of the sticks on which some markers are placed. This information is useful in determining the reliability of the calculation of the segment position and orientation.

Segment based information

Segment Parameters: Estimates of segment mass, inertia relative to the principal axes of inertia, and joint positions relative to segment center of mass based upon published experimentally derived regression equations. Segment local coordinate system is assumed to be aligned with the principal axes of inertia (the assumption alluded to previously- this may be eliminated by a more rigorous technique (Kaleps, 1984) which requires the measurement of the 3D locations of some 83 points!) and thus the products of inertia are 0. Thus, inertia is expressed as 3 vector rather than a 3x3 matrix. Separate databases are used for male, female, and children (Jensen, 1989).

Segment Position: Segment position and orientation relative to the lab global coordinate system (GCS). Expressed in terms of the position of the segment mass center and the orientation of the segment principal moment of inertia axes via a 3x3 rotation matrix. Note that since a rotation matrix is used there is no built in assumption as to a type euler angle, or whatever, system that is used to express the rotations. This is most useful for subsequent mathematical analyses, but causes difficulty in interpretation.

Average segment position: Average position and orientation of the segment in all motion data. Useful for looking at average standing posture as well as studying and correcting for biases due to marker placement on the segments.

Segment angle: The yaw-pitch-roll euler angles of each segment relative to the GCS. In this way, the angle of the foot relative to ground may be studied, for instance. This provides a method to interpret the 3x3 rotation matrix, though not the best since euler angles are rather obscure at times also.

Segment velocity and acceleration: Segment velocities and accelerations, both translational and rotational, relative to the lab GCS and the segment local coordinate system (LCS). This information is useful in interpreting kinematic data since it is dependent upon the first and second derivatives of motion and thus shows changes in motion much more clearly. These derivatives are computed using either the GCV, DPF, or finite difference methods. Note that fully three dimensional kinematic calculations are used in computing these quantities.

Segment energy: The rotational, translational, and total kinetic energies of each segment are available for each frame of motion data. Potential energy of the segment center of mass is defined relative to the GCS. Total energy of segment which is a scalar summation of the total kinetic and potential energies of the segment.

Body energy: Same as for individual segments except that they are the sum of all the kinetic, potential, and total energies of all the segments represented in the marker set.

Segment momentum: The linear and angular momentums of each segment in the segment's LCS. In addition to these vectorial quantities, the magnitudes of both momentums are also calculated.

Segment global screw: The screw axis defining the motion of the segment in the GCS from one frame to the next. This is an instantaneous screw since the position of the segment in the previous frame is used as the reference for the calculation of the screw. Note that the screw is expressed in the GCS rather than the LCS of the segment. A separate calculation of the instantaneous screw of each joint (the motion of the distal segment relative to the proximal segment) is also carried out. In addition, the finite screw axis for the same motions is also calculated for comparison/research purposes. Both the relative screw for the segments's motion form one frame to the next as well as the absolute screw for the kinematic state of the segment relative to the GCS can be calculated. Several calculation methods are present for calculating the finite screw since studies have shown that some methods are more sensitive to errors due to noisy data than are other methods [Fenton, 1989].

Joint based information

Joint angle: Joint angles are calculated relative to the joint coordinate system (JCS) and the proximal segment LCS. The JCS is a nonorthogonal coordinate system which more closely resembles the way in which orthopaedists describe joint angles than do other euler angle sequences. The JCS for each joint is slightly different and as a result they should be presented with diagrams and mathematical formulae to give each joint a precise definition. The concept of this coordinate system is based upon a paper by Suntay and Grood (1986). Their coordinate system was defined only for the knee, but here it has been generalized to the 15 joints of the body which *ANALYZE* recognizes. In addtion, standard euler yaw-pitch-roll angles have been included for completeness sake since some previous studies have used this convention. Note that the sagittal plane angles (flexion-extension) are very similar for both coordinate systems and in fact have been analyzed theoretically to show that they do not deviate greatly until a fair amount of non-sagittal angulation occurs. The non-sagittal angles (in/external, ab/adduction, pro/supination, etc.) are a great deal different from one another, however.

Average joint angles: Average joint angle over the period of the motion data. This is useful for looking at biases in the calculated joint angles resulting from the placement of the markers upon the body segments.

Joint velocity and acceleration: The joint velocities and accelerations in both the JCS and proximal segment LCS's are calculated in either of two ways. The kinematically correct way utilizing the absolute rotational velocities and accelerations relative to the lab GCS of the segments on either side of a joint or by numerically differentiating the individual joint angle components using the GCV, DPF, or finite difference methods. Note that this second technique, though used by many previous researchers, is not correct for calculating the true joint velocities and accelerations. The error is not particularly significant unless the joint motion has significant components in planes other than the sagittal which the case with pelvic motion especially and in hip motion to a lesser degree.

Joint reaction load: The joint reaction loads (forces and moments) are computed at the current joint position and segment kinematic state. Note that the joint reaction loads (i.e., 3 moments and 3 forces) are calculated since all calculations are carried out in full 3D with no assumptions as to the type of motion being studied. A number of previous software packages have limitations in this respect for either they assume sagittal plane motion or they only calculate joint moments. The loads are given relative to the JCS, the proximal segment LCS, the distal segment LCS and the lab GCS. All four are calculated to enable comparison with other previous studies since there has not been uniformity in the way in which the joint loads have been expressed in the literature published to date. The loads are always given in terms of the load exerted by the proximal segment upon the distal segment across the joint. The scheme used to calculate the joint loads is very flexible for it is based upon the concept of chained calculations. Depending upon the activity, the phase in the activity, and the degree to which the ground reaction loads are known, an appropriate joint load calculation method is used. For example, if the reaction loads under the stance leg during single stance are known then a sequence of calculations which moves from the support foot up to the hip and then starts at the swing foot and moves up to the hip of the swing leg also is used to calculate the leg joint reaction loads. The arms, neck and trunk joint reaction loads are calculated by progressing from distal to proximal across the arms and then the trunk. If, however, the ground reaction loads under the support foot are not known then a calculation sequence which progresses up the swing leg, along both arms, down the neck, then to the trunk, and then lastly down the support leg is utilized. Note that the resulting calculated joint reaction loads in the support leg will, without a doubt, be different from those calculated using the ground reaction if it were known. This is primarily due to inadequate determination of the mass/inertia properties and joint locations of the segments further up the calculation chain. Any inaccuracies will be cumulative due to chaining of the calculations, however at least an estimate is available whereas normally none would be calculated in other software packages. The calculation scheme will change throughout the data set depending upon what the round contact/valid force plate data combination is at a given data frame. A number of schemes have been developed for full body joint reaction load calculation. To do only upper body for example, a connectivity structure, which guides the calculation sequence, would have to be defined within *ANALYZE*. Lastly, if at any frame of motion data the reaction loads of a joint cannot be calculated (the joints of both legs during double stance of human gait when neither foot's ground reaction load is known), then an array indicating the validity of a joint's reaction load calculation is updated and later in *TELIO* this data is automatically skipped over when it is graphed or displayed in 3D since it is invalid.

Support load: The vectorial summation of the joint reaction loads of the joints (ankle,knee,hip) in the individual legs. This is of interest in studying the process of gait (See Winter, 1987).

Joint reaction wrench: The joint reaction load may also be expressed in terms of a load wrench, which like its kinematic analog, the instantaneous screw, is an invariant quantity in terms of the coordinate system in which it is expressed. This may prove useful as a way to compare the joint reaction loads calculated by different programs and/or labs. It may also be useful in interpreting the joint reaction loads in a more succinct and consistent manner. This is needed at times because the resulting joint reaction load patterns in 3D can be quite complex. Previous studies have looked primarily at the motion moments (saggital plane joint moments consisting of both active and passive moments) and have ignored the structural moments (such as the ab/adduction resultant moments and an/posterior force in the knee). These may prove to be important in areas such as compensation of gait where a patient may be trying to minimize the structural moment to protect a specific structure (e.g., minimizing abduction moments at the knee to minimize strain on the medial collateral ligament). Anyway, the capability is here for the using.

Joint screw: The screw defining the motion of the distal segment relative to the proximal segment across the joint from one motion frame to the next. This is an instantaneous screw using the positions and velocities of the distal and proximal segments. Note that the distal segment position/velocity is computed in the proximal segment LCS in each frame in the process of the screw calculation, so even though the proximal segment may be moving the screw is still calculated properly. This screw defines the relative motion of the segments across the joint. The literature indicates that the calculation of screws is very sensitive to errors in position data and thus unless the segment position and orientation is determined using redundant markers (more than the minimum of 3 per segment), the resulting screws may not be very useful. I have not yet had a chance to check this out. This calculation could also be applied to data gathered from an instrumented spatial linkage (ISL) (another future project...) which would presumably give better results. Lastly, finite screw calculations have been included for comparison/research purposes.

Joint Screw Angle: Essentially the direction vector of the finite screw axis of the joint multiplied by the rotation angle about the axis and projected onto the proximal segment, distal segment, or JCS coordinate systems.

Joint Translation: The translation across the joint based upon the assumption that joint position relative to the proximal and distal coordinate systems is fixed. If not then the point indicated by the two segment LCS's is not the same and hence the difference between the two points is the calculated joint translation. At present, this is used as a diagnostic measure to check the assumption that the ankle, knee, hip, elbow, and shoulder markers are being placed near the axis of rotation of the joint.

Joint power transfer: The flow of power across the joints is calculated using the joint rotational velocities and the joint reaction moments. This transfer consists of both active (contracting muscles) and passive (resistance of the ligaments, capsule, passive elasticity of the muscles) power lumped into a single quantity.

Joint Reaction Load Moment Arms: The moment arms of each joint reaction loadmoment component relative to the joint location is calculated using the joint reaction forces in the other planes. For instance, the moment arm of the x moment component is calculated using the y and z components of the resultant joint force.

Joint Wrench Arm: The perpendicular distance from the joint reaction load wrench axis to the joint location is calculated.

Force Plate Information

Individual and combined foot ground reaction load: The force place data is summed and split such that the loads under each individual foot and their combination is computed. The only time these are different is during the double support phase of gait.

Individual and combined foot centers of pressure: The center of pressure (COP) of the ground reaction loads measured under each foot as well as the center of pressure from the combined feet. The only time these two are different is during the double support phase of gait.

Center of mass via integration: The location of the center of mass in 3D of the entire body is calculated by integrating the ground reaction force twice and scaling it by the body mass. This may be compared directly with the center of mass calculated using the estimated segment masses and center of mass locations.

Ground reaction wrench: Wrench of the ground reaction load for the individual feet and the combined foot loads.

Center of pressure relative to foot LCS: Position of the ground reaction COP during the contact phase of each foot relative to the foot' s LCS.

Strike index: The position of the COP relative to the total anterior/posterior distance along the foot and the position of the foot's LCS. This requires that the foot length be calculated. At present *ANALYZE* uses the toe and heel markers of the whole body marker set to do this calculation.

Individual and combined foot ground reaction impulse: The impulse of the foot upon the ground.

Reconstructed ground reactions: Using the segment accelerations combined with their mass and inertia as well as geometric configuration, the ground reaction loads, both magnitude and location, are calculated.

Pitching moment arm: The perpendicular distance from the line formed by the ground reaction force passing through the center of pressure to the location of the body center of mass. This can be thought of as a rough estimate of what makes the body pitch forward/backward, side to side, and twist.

Wrench pitching arm: The perpendicular distance from the line formed by the ground reaction load wrench axis to the location of the body center of mass.

EMG's

Filtering: Digital filtering with high, low, and notch filter capabilites.

Rectification: Rectify the signal

Integration: Integrate the signal

Moving Window Average: Perform a moving window average with a user specified window size.

Threhold: When combined with several of the above signal processing operations this can provide a way of displaying when a muscle is or is not active in an on/off fashion.

Comment on numerical capabilities

At present *ANALYZE* contains three different ways to filter data. One method is a generalized cross validation routine utilizing splines developed by Herman Woltring which is configurable from within *ANALYZE*. Another generalized cross validation routine from Henry Busby of The Ohio State University which does not use splines is also included. Lastly, a general digital filter routine can be used in which its upper and lower cutoff frequencies can be set as well as have a notch defined. This routine can have the slope of the cutoffs and notch specified so that the user may create their own specific filter functions. It also contains a Hanning window function which may be used or turned off. Any of these filter operations may be chained if so desired. Several methods have been provided because experiments have shown merit and inadequacies in all of them. It was felt that the user should have final say on what is used by being able to experiment with their data since it may have unique properties which make it work better with one method or another. Most quantities within *ANALYZE* are filterable.

How is ANALYZE presently implemented ?

In its present implementation, *ANALYZE* is not being used optimally, but at the same time practical, useful results are being generated. Currently, *ANALYZE* runs on a whole body marker set consisting of 21 markers, an upper body marker set consisting of 9 markers which includes the head, trunk, pelvis, right upper arm, and right lower arm, and a simple 6 marker set for measuring foot motion only. The full body marker set does not allow the determination of true 3D position and orientation of some of the segments because in some cases markers are shared by two segments. In these cases, the markers are assumed to sit at the joint "center", or on the axis of rotation, and thus should be sharable between segments since the common center would not move relative to the two segment LCS's. A primary reason for this marker set being used is that it has proved to be a very practical arrangement for day to day, and often several times a day, use upon patients with various gait pathologies. Since the marker data is being collected upon a VICON system, minimization of technician time in sorting and marker identification errors is very important. With the introduction of AMASS, the new 3D marker trajectory reconstruction software of Adtech and distributed by Oxford Metrics, this point is becoming less important. The marker set is an effective compromise between measurement of motion which is clinically useful yet still can be used upon all types of pathological motions without the markers being confused. Machine independency of code and data files has been extensively tested by producing a version of *ANALYZE* on the Macintosh using the same FORTRAN source code used for the VAX version. In addition, data files produced by the two programs have been interchanged successfully.

Marker data can presently be read in from VICON TR3 files, AMASS C3D files, and an older file format used at Ohio State called TRU files. Force plate data is read in from C3D and FPD files as well as from from two file formats used at Ohio State- GLS and GLS2 files. The results can be saved in either *ANALYZE* format (.ANZ) or the older TRU format. Results can also be dumped in text form to allow export to other packages such as statistics and presentation graphics. The number of file formats supported will be expanded in the future depending upon how widely *ANALYZE* is used.

The Future

At present, the public release of ANALYZE is projected for the fall of 1991. Full documentation including derivation of the algorithms used in the FORTRAN code as well as a user's guide will be available. The source code will also be made available. The only charge will be to cover the costs of reproduction. The intention in releasing ANALYZE, as well as TELIO, in its entirety is to provide other research groups with a resource which may be studied, used and/or modified for their specific needs. As such, the packages are not intended to be a commercial product and thus there will be no support. It will be user supported software— you use it, you support it!

NOTE: This project is intended to make 3D motion analysis capabilities widely available to all human motion researchers. It is not to be used by commercial ventures to generate a product loosely based upon this work and then charge ridiculous fees for the resulting application. Thus, legalise will be included in the release version which will hopefully allow those for whom it is intended to use it freely and yet will prevent those who represent the antithesis of this project from obtaining financial gain.

References

1 Busby, H. R. and D. M. Trujillo (1985). *Numerical Experiments with a Differentiation Filter.* J Biomech Eng 107: 293-299.

2 Dohrmann, C. R., H. R. Busby, et al. (1988). *Smoothing Noisy Data using Dynamic Programming and Generalized Cross-Validation.* J. Biomech. Eng. 110: 37-41.

3 Fenton, R. G. and X. Shi (1989). *Comparison of Methods for Determining Screw Parameters of Finite Rigid Body Motion from Initial and Final Position data.* Advances in Design Automation-1989, Montreal,Quebec, ASME, 433-439.

4 Grood, E. S. and W. J. Suntay (1983). *A Joint Coordinate System for the Clinical Description of Three-Dimensional Motions: Application to the Knee.* J Biomech Eng 105: 136-144.

5 Jensen, R. K. (1989). *Changes in Segment Inertia Proportions Between 4 and 20 Years.* J Biomech 22(6/7): 529-536.

6 Kaleps, I., C. E. Clauser, et al. (1984). *Investigation into the Mass Distribution Properties of the Human Body and its Segments.* Ergonomics 27(12): 1225-1237.

7 McConville, J. T., T. D. Churchill, et al. (1980). *Anthropometric Relationships of Body and Body Segment Moments of Inertia.* United States Air Force Aerospace Medical Research Laboratory, AFAMRL-TR-80-119.

8 Woltring, H. J. (1986). *A Fortran package for generalized, Cross-validatory Spline Smoothing and Differentiation.* Adv. Eng. Software 8(2): 104-113.

9 Young, J. M., R. F. Chandler, et al. (1983). *Anthropometric and Mass Distribution Characteristics of the Adult Female.* Federal Aviation Administration, FAA-AM-83-16.

Some Notes on Kinematics Calculations in ANZ

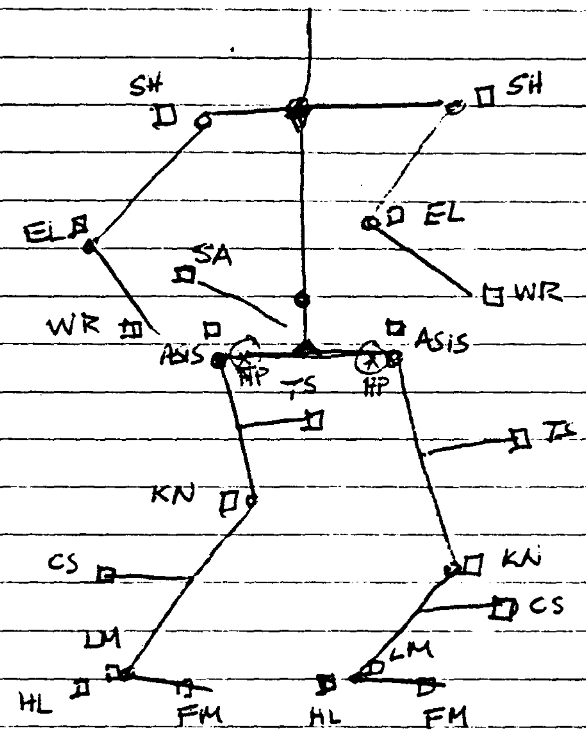
Position/Orientation calculation

The transformation from a time series of marker positions to a time series of segment positions and orientations is carried out in a simple manner in ANZ. Various combinations of three markers is used to calculate these segment kinematics. There are notes describing the vector cross product order and other operations used to do these calculations. The result of doing the transformations in this manner is that the locating of the segment local coordinate systems is very dependent upon the placement of the markers. In addition, since only three markers per segment is used, the calculations are not as accurate as they could be if redundant markers were placed on the segments. Algorithms to take advantage of such marker sets, such as the Spoor & Veldpaus or the modified Schut algorithm used by Antonsson, are not implemented in ANZ. I am working on a more general spatial kinematics package, written in C, which incorporates four different algorithms to do this calculation (so that they can be used to cross check one another). This will not be done for a while however. In the meantime, the calculations used here are adequate for clinical locomotion studies, but they should not be applied to detailed kinematic evaluations such as spatial kinematics of a joint like the knee. The methods used here are very similar to those used in Helen Hayes marker set as well as those incorporated into Motion Analysis Corporation's Orthotrak software package.

Rotational and translational velocity and acceleration

The rotational and translational velocities and accelerations are computed using spatial kinematic formulations. Translational vel/accel is a simple matter. The computation of the rotational vel/accel is another matter. Three different algorithms have been programmed to allow cross checking of the rotational vel/accel calculations. The three methods are: 1) Direct differentiation of the rotation matrix elements, 2) differentiation of the segment euler angles, 3) differentiation of the marker trajectories. The first two methods have notes about their derivation in the ANZ documentation, although both are common approaches described in kinematics texts. The marker based approach is based upon a paper by Verstraete. In all cases the differentiation of quantities is usually carried out using the GCVSPL routine of Herman Woltring although it can be overridden and finite difference used instead for comparison purposes.

OSU/Boston Marker Set



HP locations are computed from ASIS & KN markers using the method published by Tylkowski & Simon (See comments in the source code for the Adjust Marker Data routine in marker.for)

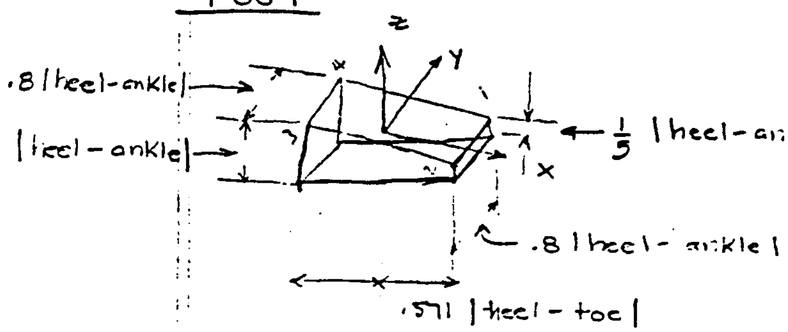
□ HL - heel
LM - lateral mallealous (adjusted)
FM - First metatarsal
CS - Calf stick
KN - Knee (lateral) (adjusted)
TS - Thigh stick
AS - ASIS
SA - Sacral
EL - elbow
SH - shoulder
WR - wrist

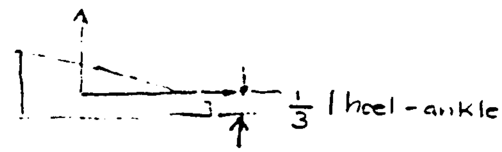
⊛ HP - hip (computed)

Sizing of segment parallelapipeds
- based upon my measurements
- expressed as ratio of length of long axis of segment

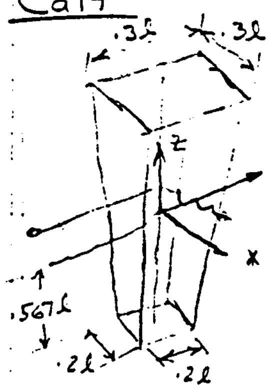
(These are used only for making Shao3d views)

Foot





Calf

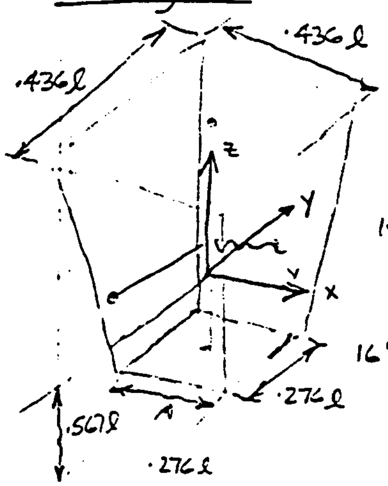


16½" = $2\pi r_k$ d_k = 5.252 K_k = .309

17"

11" = $2\pi r_a$ d_a = 3.501 K_a = .206

Thigh

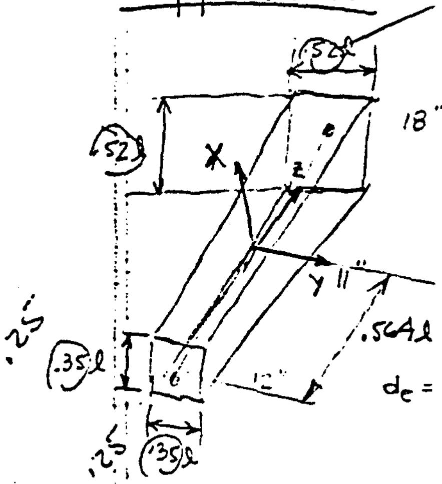


26" d_h = 8.276 K_h = .436

19"

16½" d_k = 5.252 K_k = .276

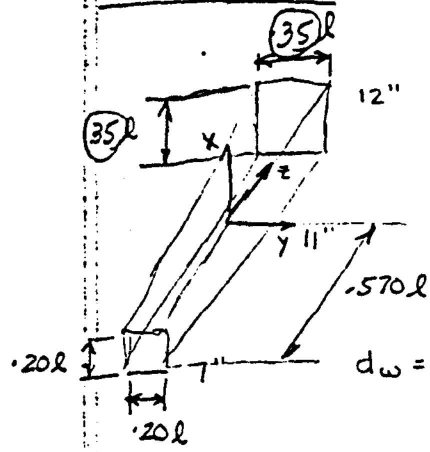
Upper Arm



18" $d_s = 5.780$ $K_s = .520$

$d_e = 3.825$ $K_e = .350$

Lower Arm

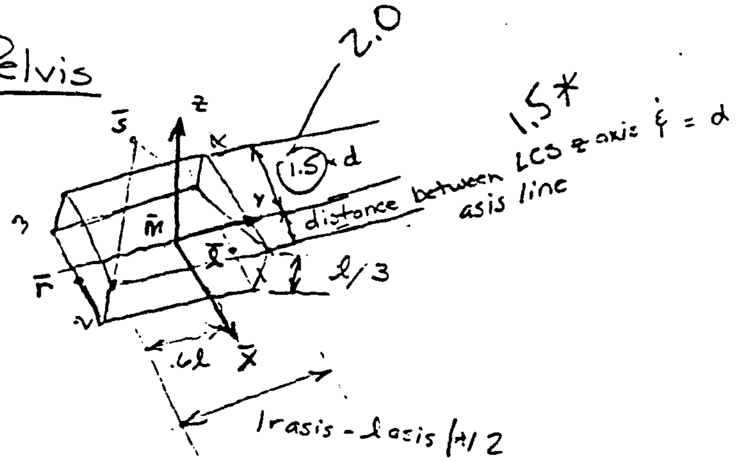


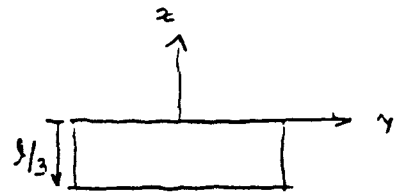
12" $d_e = 3.820$ $K_e = .350$

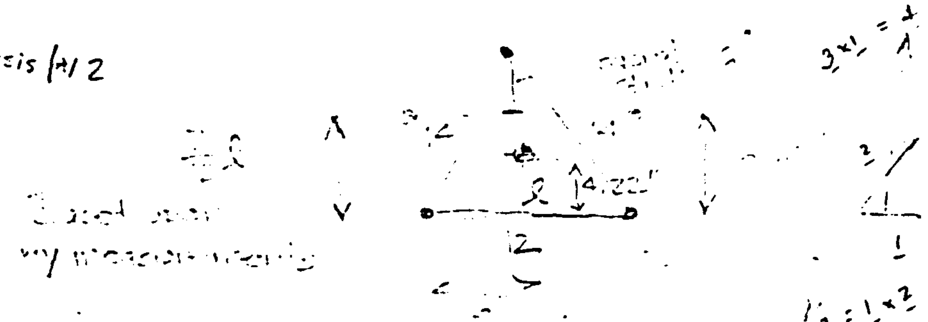
$d_\omega = 2.228$ $K_\omega = .200$

$$\bar{m} = \frac{\frac{\bar{r}+\bar{l}}{2} + (\bar{s} + L\bar{x})}{2}$$

Pelvis

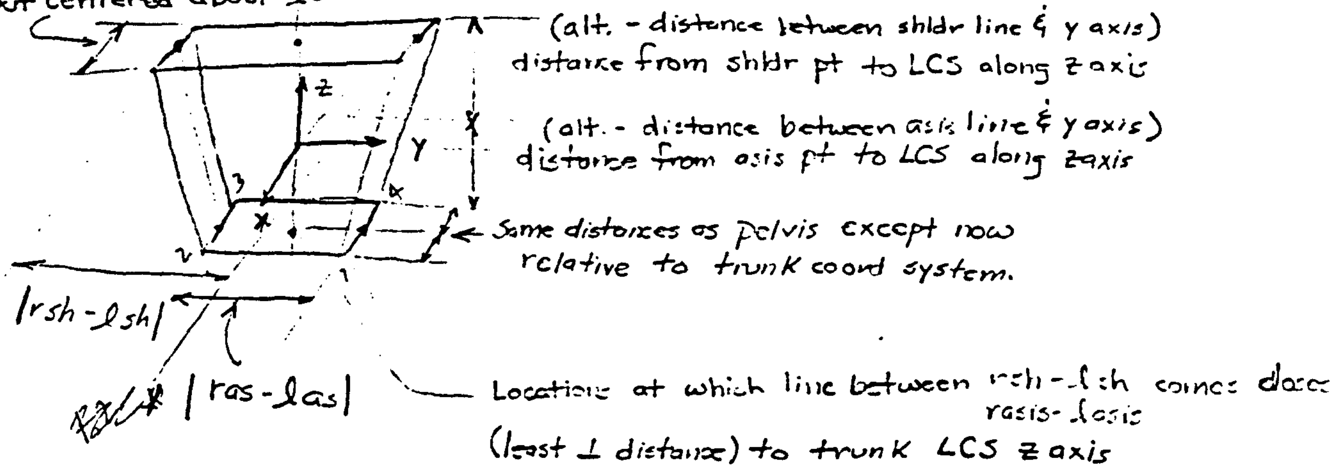




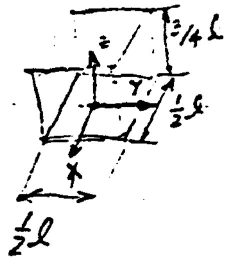


Trunk

Same width as at pelvis but centered about lsh-rsh line



Head



ℓ = |rsh - lsh|

1-24-89

Connectivity of Body in <u>Analyze</u>

Angle (Joint) #	Name	Seg. 1 #	Name	Seg. 2 #	Name
1	rtak	1	~~rtft~~	2	rtcf
2	rtkn	2	~~rtcf~~	3	rtth
3	rthp	3	rtth	7	pelv
4	lfak	4	lfft	5	lfcf
5	lfkn	5	lfcf	6	lfth
6	lfhp	6	lfth	7	pelv
7	pvlb	7	pelv	0	lab
8	pvtk	8	trnk	7	pelv
9	rtel	10	rtla	9	rtua
10	rtsh	9	rtua	8	trnk
11	lfel	12	lfla	11	lfua
12	lfsh	11	lfua	8	trnk
13	neck	13	head	8	trnk
14	rtwr	14	rthd	10	rtla
15	lfwr	15	lfhd	12	lfla

Angles are defined as Seg. 1 moving relative to Seg. 2

this data is in the array jnt-seg-num(2,[illegible]

Connectivity of body for various combinations of ground contact conditions

Data specified in each case:

- # of jnts
- jnt load calculation order
- Seg. LCS in which jnt load calc's are performed
- # of jnts connected to seg whose jnt load is being calc'c
- jnts #'s connected to Seg.
- # ext. loads applied to Seg. — Provision for extra externally applied loads - such as a cane!
- ext. load id #'s

No Ground Reaction Known, Swing phase — TOP DOWN

Rt. stance leg JNT_TD(rt)

12

neck	rtel	rtsh	lfel	lfsh	trnk/plv	lfak	lf kn	lfhp	rthp	rtkn	rtak
head	rtla	rtua	lfla	lfua	trnk	lfft	lfcf	lfth	plv	rtth	rtcf
0	0	1	0	1	3	0	1	1	2	1	1
—	—	rtel	—	lfel	neck rtsh lfsh	—	lfak	lfkn	lfhp trnk/plv	rthp	rtkn

0 ⟶

⟶

Lf stance leg JNT_TD(lf)

12

neck	rtel	rtsh	lfel	lfsh	pvtk	rtak	rtkn	rthp	lfhp	lfkn	lfak
head	rtla	rtua	lfla	lfua	trnk	rtft	rtcf	rtth	pelv	lfth	lfc
0	0	1	0	1	3	0	1	1	2	1	1
—	—	rtel	—	lfel	neck rtsh lfsh	—	rtak	rtkn	rthp pvtk	lfhp	lf

0 ⟶

0 ⟶

Ground Reaction Known – Swing Phase – BOTTOM UP

Rt. Stance Leg   JNT_BU(rt)

#jnt	12											
jnt_order	neck	rtel	rtsh	lfel	lfsh	prtk	lfak	lf Kn	lfhp	rtak	rtKn	rthp
jnt seg LCS	head	rtla	rtua	lfla	lfua	trnk	lfft	lfcf	lfth	rtft	rtcf	rtth
jnt-seg	0	0	1	0	1	3	0	1	1	0	1	1
jnt #'s	—	—	rtel	—	lfel	neck rtsh lfsh	—	lfak	lfKn	—	rtak	rtkn
#ext load	0	0	0	0	0	0	0	0	0	1	0	0
ext load #	0	0	0	0	0	0	0	0	0	1	0	0

Lf. Stance Leg   JNT-BU(lf)

Same as above except for ext loads

#ext load	0	0	0	0	0	0	ø	0	0	0	0	0
ext load #	0	0	0	0	0	0	2	0	0	0	0	0

Ground Reaction Known (Both feet) – Double Stance   JNT-DBL(3)

Same as above except for ext loads

#ext load	0	0	0	0	0	0	1	0	0	1	0	0
ext load #	0	0	0	0	0	0	2	0	0	1	0	0

Ground Known for one foot – Double Stance

Rt. foot Known JNT-DBL(rt)

#jnt	12											
jnt-order	neck	rtel	rtsh	lfel	lfsh	pvtk	rtak	rtkn	rthp	lfhp	lfkn	lfak
jntseg LCS	head	rtla	rtua	lfla	lfua	trnk	rtft	rtcf	rtth	pelv	lfth	lfcf
jnt-seg	0	0	1	0	1	3	0	1	1	2	1	1
jnt #'s	—	—	rtel	—	lfel	neck rtsh lfsh	—	rtak	rtkn	~~[illegible]~~ pvtk rthp	lfhp	lfkn
ext loads	0	0	0	0	0	0	1	0	0	0	0	0
xf load#	0	0	0	0	0	0	1	0	0	0	0	0

Lf foot Known JNT-DBL(lf)

12											
neck	rtel	rtsh	lfel	lfsh	pvtk	lfak	lfkn	lfhp	rthp	rtkn	rtak
head	rtla	rtua	lfla	lfua	trnk	lfft	lfcf	lfth	pelv	rtth	rtcf
0	0	1	0	1	3	0	1	1	2	1	1
—	—	rtel	—	lfel	neck rtsh lfsh	—	lfak	lfkn	lfhp pelv	rthp [illegible]	rtkn
0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	2	0	0	0	0	0

Ground Reaction Unknown – Double Stance

JNT-DBL(4)

6					
neck	rtel	rtsh	lfel	lfsh	pvtk
head	rtla	rtua	lfla	lfua	trnk
0	0	1	0	1	3
—	—	rtel	—	lfel	neck rtsh lfsh

0 ⟶

0 ⟶

NOTE: JNT_DBL (5) & (6) Added 11/21/89

(5): rt. only but don't eval. lf leg

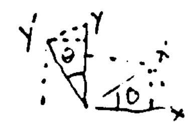
(6): lf. only but don't eval. rt leg

Floating Axis Euler Angles (JCS)

Order $\theta - \phi - \psi$ ← flexion - abduction - internal

$$R = R_Y(\theta)\, R_X(\phi)\, R_Z(\psi)$$

Same conventional as Grood & Suntay, Chao, etc...



$y' = -xs\theta, yc\theta$

$x' = xc\theta, ys\theta$

$\begin{Bmatrix} x' \\ y' \end{Bmatrix} =$

$$= \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix} \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} c\psi & -s\psi & 0 \\ c\phi s\psi & c\phi s\psi & -s\phi \\ s\phi s\psi & s\phi c\psi & c\phi \end{bmatrix}$$

$$= \begin{bmatrix} c\theta c\psi + s\theta s\phi s\psi & -c\theta s\psi + s\theta s\phi c\psi & s\theta c\phi \\ c\phi s\psi & c\phi c\psi & -s\phi \\ -s\theta c\psi + c\theta s\phi s\psi & s\theta s\psi + c\theta s\phi c\psi & c\theta c\phi \end{bmatrix}$$

$$c\phi = \sqrt{r_{21}^2 + r_{22}^2}$$

$$\theta = \text{ATAN2}(r_{13}/c\phi,\ r_{33}/c\phi)$$

$$\psi = \text{ATAN2}(r_{21}/c\phi,\ r_{22}/c\phi)$$

$$\phi = \text{ATAN2}(-r_{23},\ c\phi)$$

if $c\phi = 0$ (or error)

if $r_{23} < 0$

$\phi = 90°$ $\quad \psi = 0°$ $\quad \theta = \text{ATAN2}(r_{12}, r_{11})$

$$\begin{bmatrix} c\theta & s\theta & 0 \\ 0 & 0 & -1 \\ -s\theta & c\theta & 0 \end{bmatrix}$$

else if $r_{23} > 0$

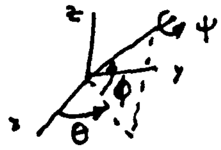
$\phi = -90°$ $\quad \psi = 0°$ $\quad \theta = \text{ATAN2}(-r_{12}, r_{11})$

$$\begin{bmatrix} c\theta & -s\theta & 0 \\ 0 & 0 & 1 \\ -s\theta & -c\theta & \end{bmatrix}$$

Used with (Shoulder) in upper body marker set.

Polar (Z Y Z) Euler Angles

Azimuth - Inclination - Twist



$$R = R_z(\theta)\, R_y(\phi)\, R_z(\psi)$$

$$= \begin{bmatrix} c\theta c\phi c\psi - s\theta s\psi & -c\theta c\phi s\psi - s\theta c\psi & c\theta s\phi \\ s\theta c\phi c\psi + c\theta s\psi & -s\theta c\phi s\psi + c\theta c\psi & s\theta s\phi \\ -s\phi c\psi & s\phi s\psi & c\phi \end{bmatrix}$$

$$s\phi = \sqrt{r_{13}^2 + r_{23}^2}$$

$$\theta = \mathrm{ATAN2}\left(r_{23}/s\phi,\ \cdot r_{13}/s\phi\right)$$

$$\psi = \mathrm{ATAN2}\left(r_{32}/s\phi,\ -r_{31}/s\phi\right)$$

$$\phi = \mathrm{ATAN2}\left(s\phi,\ r_{33}\right)$$

if $s\phi = 0 \longrightarrow$ (or error)

if $r_{33} > 0$ then

$\phi = 0°$

$\psi = 0°$

$\theta = \mathrm{ATAN2}(-r_{12}, r_{11})$

$$\begin{bmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

else if $r_{33} < 0$ then

$\phi = 180°$

$\psi = 0°$

$\theta = \mathrm{ATAN2}(-r_{12}, r_{22})$

$$\begin{bmatrix} -c\theta & -s\theta & 0 \\ -s\theta & c\theta & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

...

Fixed Euler Angles - Roll, Pitch, Yaw

order $\gamma - \beta - \alpha$

Old method used to compute joint angles at Boston Children Gait Lab - not used normally in ANZ. Included for compatibility only!

$$R = R_z(\alpha)\, R_x(\beta)\, R_y(\gamma)$$

$$= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\beta & -s\beta \\ 0 & s\beta & c\beta \end{bmatrix} \begin{bmatrix} c\gamma & 0 & s\gamma \\ 0 & 1 & 0 \\ -s\gamma & 0 & c\gamma \end{bmatrix}$$

$$= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\gamma & 0 & s\gamma \\ s\beta s\gamma & c\beta & -s\beta c\gamma \\ -c\beta s\gamma & s\beta & c\beta c\gamma \end{bmatrix}$$

$$= \begin{bmatrix} c\alpha c\gamma - s\alpha s\beta s\gamma & -s\alpha c\beta & c\alpha s\gamma + s\alpha s\beta c\gamma \\ s\alpha c\gamma + c\alpha s\beta s\gamma & c\alpha c\beta & s\alpha s\gamma - c\alpha s\beta c\gamma \\ -c\beta s\gamma & s\beta & c\beta c\gamma \end{bmatrix}$$

$$c\beta = \sqrt{r_{12}^2 + r_{22}}$$

$$\alpha = \text{ATAN2}(-r_{12}/c\beta\, ,\ r_{22}/c\beta)$$

$$\beta = \text{ATAN2}(r_{32}\, ,\ c\beta)$$

$$\gamma = \text{ATAN2}(-r_{31}/c\beta\, ,\ r_{33}/c\beta)$$

if $c\beta = 0$ or (error)

if $r_{32} > 0$

$\beta = 90°$

$\alpha = 0°$

$\gamma = \text{ATAN2}(r_{13}, r_{11})$

$$\begin{bmatrix} c\gamma & 0 & s\gamma \\ s\gamma & 0 & -c\gamma \\ 0 & 1 & 0 \end{bmatrix}$$

else if $r_{32} < 0$

$\beta = -90°$

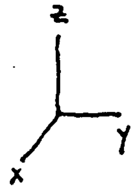
$\alpha = 0°$

$$\begin{bmatrix} c\gamma & 0 & s\gamma \\ -s\gamma & 0 & c\gamma \\ 0 & -1 & 0 \end{bmatrix}$$

$\gamma = \text{ATAN2}(r_{13}, r_{11})$

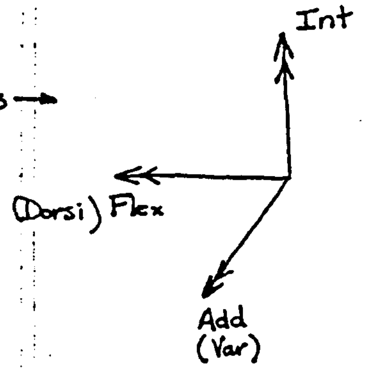
JCS Axis Definitions for all joints
(Arrow shown is positive direction)
Primarily to compute Joint Loads along JCS axes

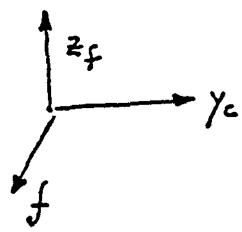
Right | Left

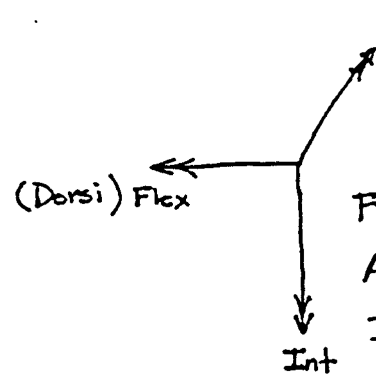


Ankle

Moments →



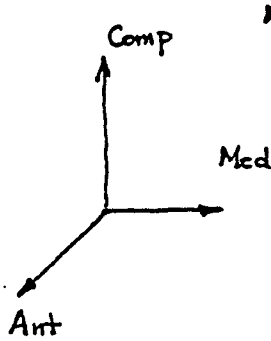


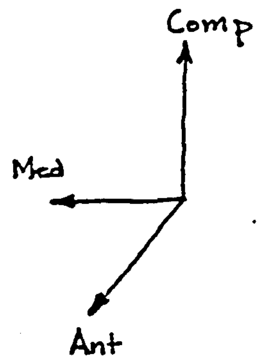


Sign correction for Joint Loads

	Rt	Lf
Flex	$-y_c$	$-y_c$
Add	$+f$	$-f$
Int	$+z_f$	$-z_f$

Forces →

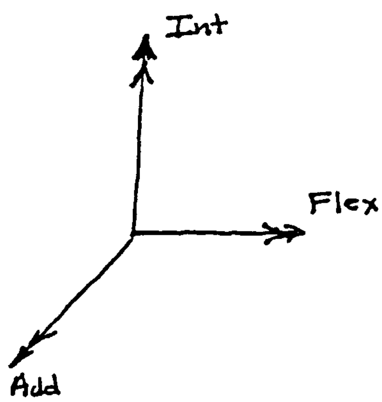


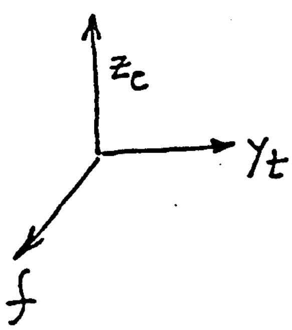


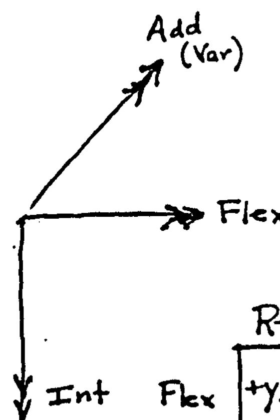
	Rt	Lf
Med	$+y_c$	$-y_c$
Ant	$+f$	$+f$
Comp	$+z_f$	$+z_f$

Knee

Moments →

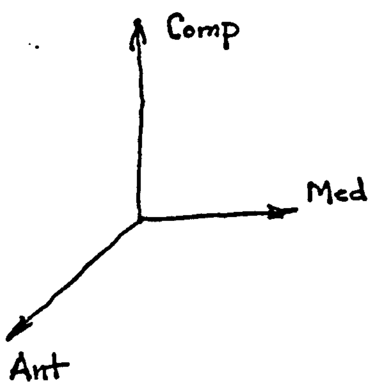


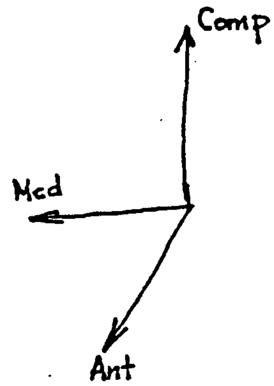




	Rt	Lf
Flex	$+y_t$	$+y_t$
Add	$+f$	$-f$
Int	$+z_c$	$-z_c$

Forces →



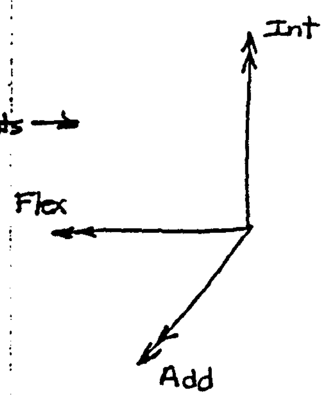


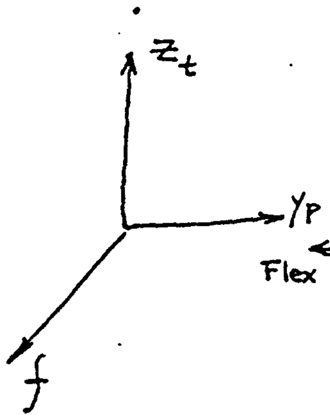
	Rt	Lf
Med	$+y_t$	$-y_t$
Ant	$+f$	$+f$
Comp	$+z_c$	$+z$

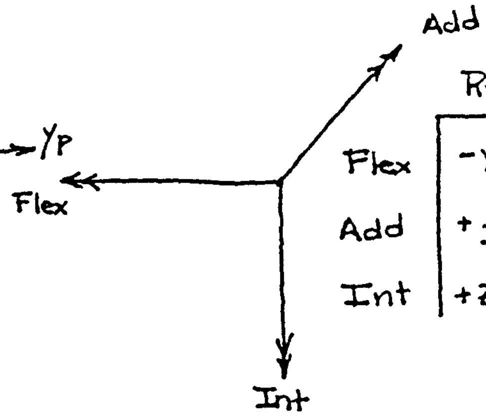
Right | Left

Hip

Moments →

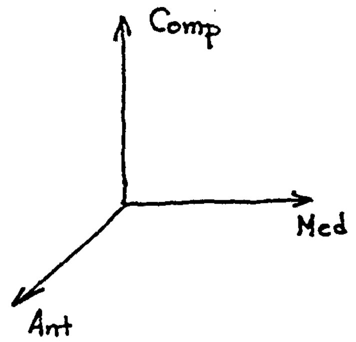


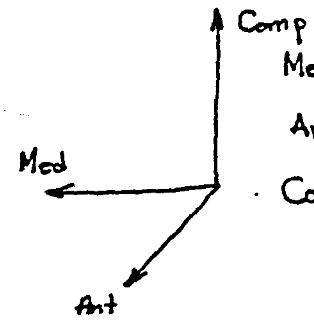




	Rt	Lf
Flex	$-y_p$	$-y_p$
Add	$+f$	$-f$
Int	$+z_t$	$-z_t$

Forces →

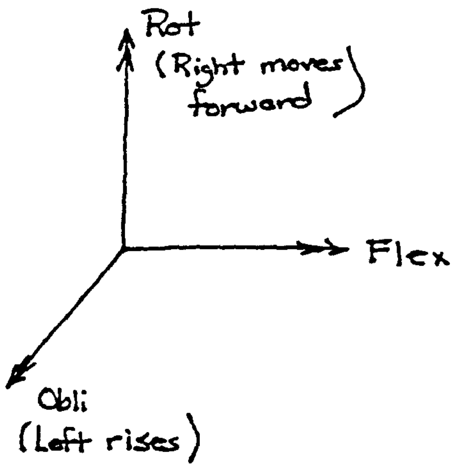




	Rt	Lf
Med	$+y_p$	$-y_p$
Ant	$+f$	$+f$
Comp	$+z_t$	$+z_t$

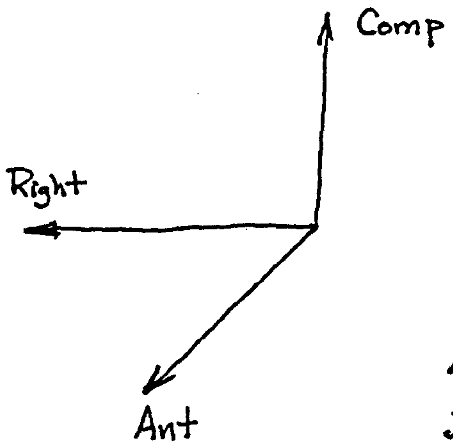
Pelvis/Lab

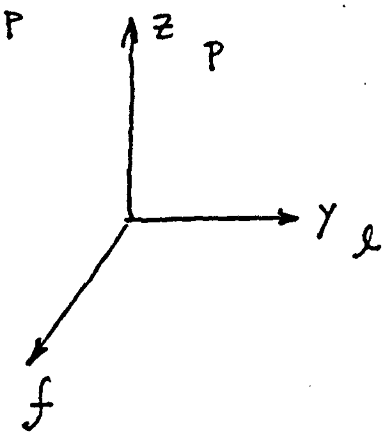
Moments →



Flex	$+y_l$
Obli	$+f$
Rot	$+z_p$

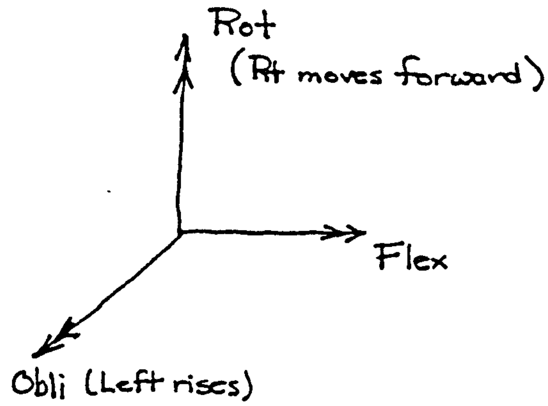
Forces →



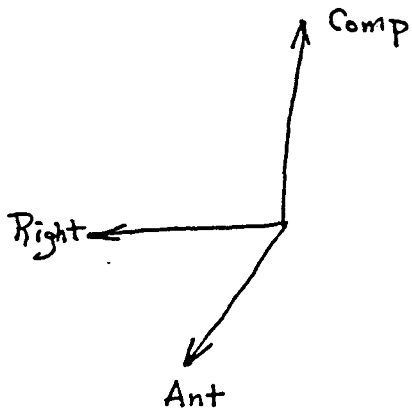


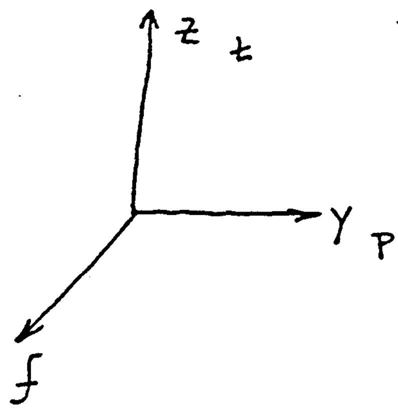
Right/Left	$-y_l$
Ant	$+f$
Comp	$+z_p$

runk/Pelvis



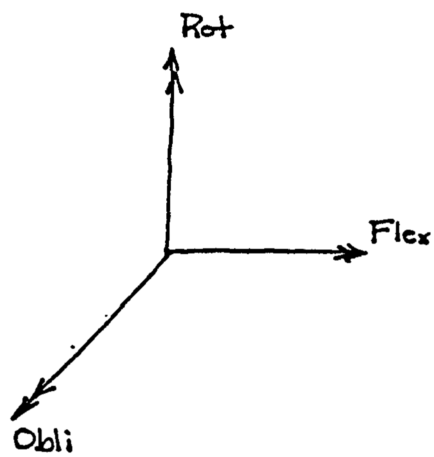
Flex	$+y_p$
Obli	$+f$
Rot	$+z_t$





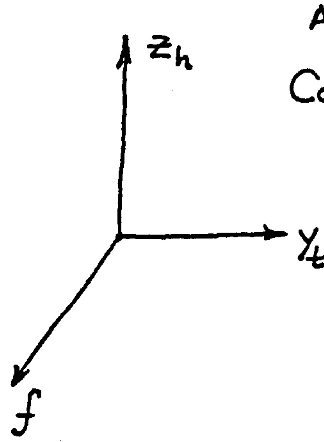
Right	$-y_p$
Ant	$+f$
Comp	$+z_t$

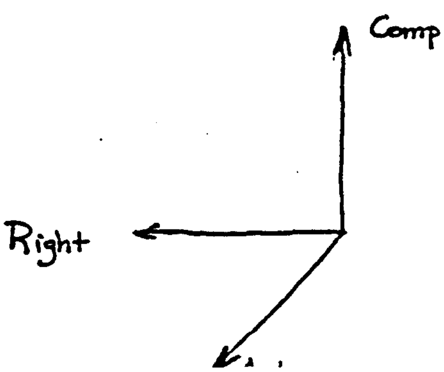
Neck



Flex	$+y_t$
Obli	$+f$
Rot	$+z_h$

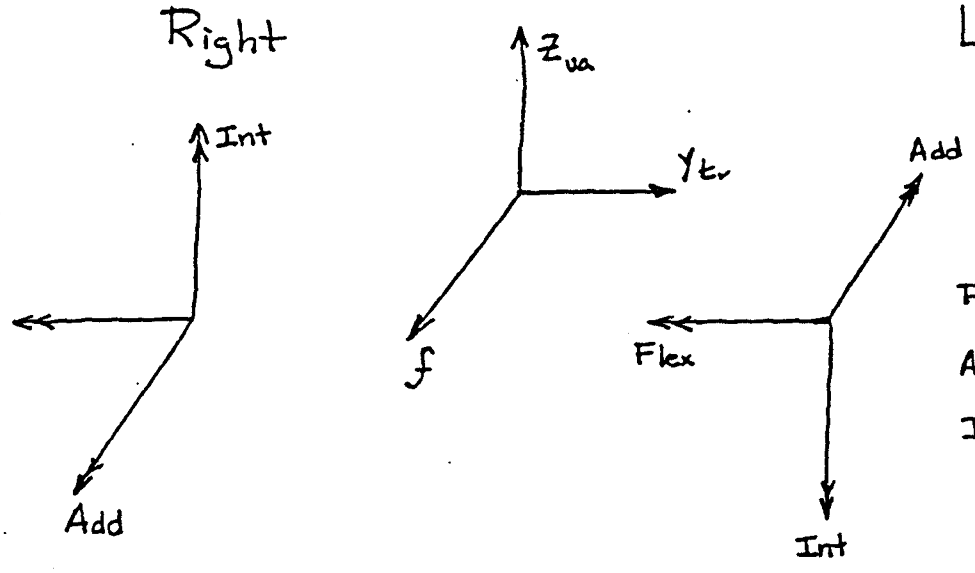
Right	$-y_t$
Ant	$+f$
Comp	$+z_h$





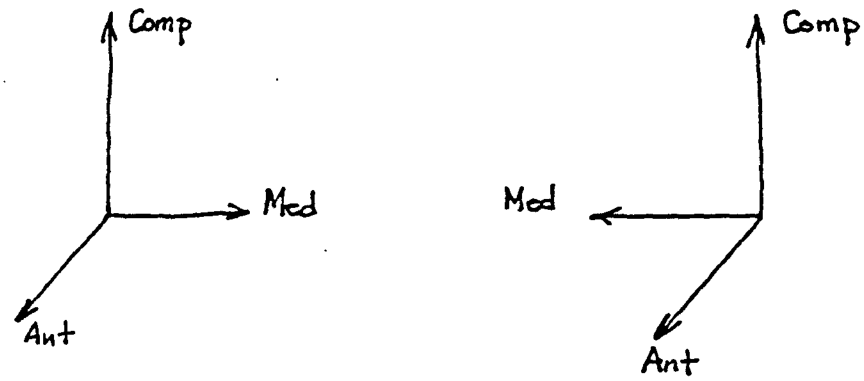
houlder

Moments →



	Rt	Lf
Flex	$-Y_{tr}$	$-Y_{tr}$
Add	$+f$	$-f$
Int	$+Z_{ua}$	$-Z_{ua}$

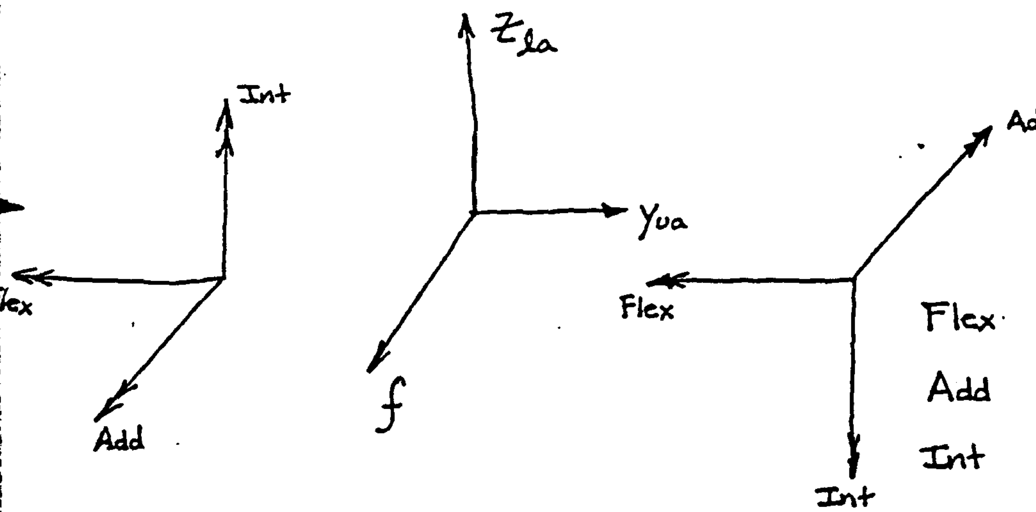
Forces →



	Rt	Lf
Med	$+Y_{tr}$	$-Y_{tr}$
Ant	$+f$	$+f$
Comp	$+Z_{ua}$	$+Z_{ua}$

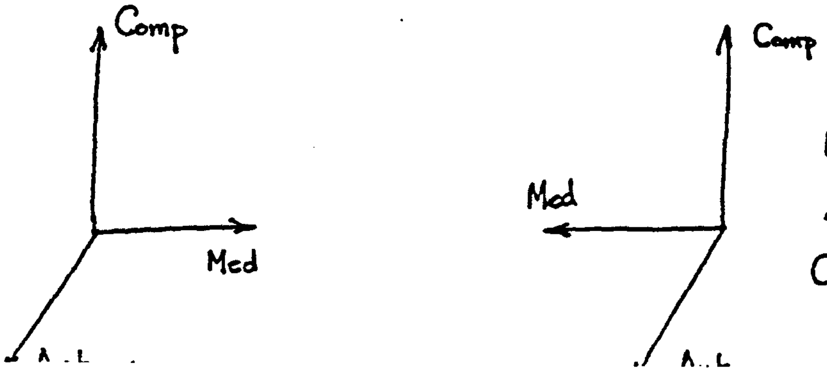
bow

Moments →



	Rt	Lf
Flex	$-Y_{ua}$	$-Y_{ua}$
Add	$+f$	$+f$
Int	$+Z_{la}$	$-Z_{la}$

Forces →



	Rt	Lf
Med	$+Y_{ua}$	$-Y_{ua}$
Ant	$+f$	$+f$
Comp	$+Z_{la}$	$+Z_{la}$

Reconstruction of Ground-Reaction Load

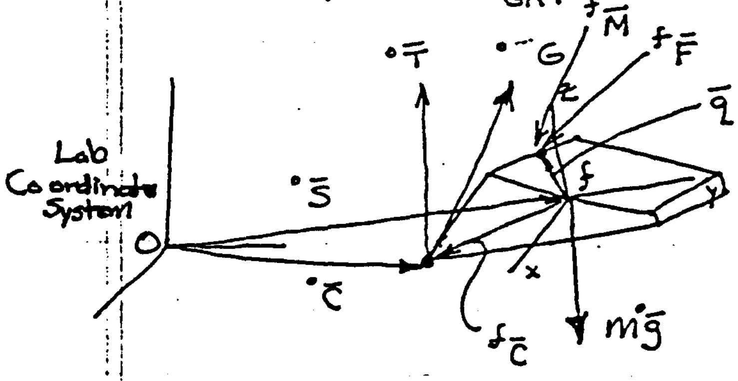
[To be used as a check when a leg's joint loads in double support are computed from top down because foot is not on a force plate & opposite foot is on a plate or in the case of single support but no force data]

Known: Ankle Resultant load & its position relative foot LCS

Foot, mass, inertia, velocity, acceleration expressed in foot's CS

Find: ${}^{o}\bar{G} = {}^{o}\begin{Bmatrix} G_x \\ G_y \\ G_z \end{Bmatrix}$; ${}^{o}\bar{T} = {}^{o}\begin{Bmatrix} 0 \\ 0 \\ T_z \end{Bmatrix}$, ${}^{o}\bar{C} = {}^{o}\begin{Bmatrix} C_x \\ C_y \\ C_z \end{Bmatrix}$ all expressed in Global CS

GRF GRT Contact Point



$^{f}\bar{G} = {}^{o}_{f}R^{T}\,{}^{o}\bar{G}$

$^{f}\bar{T} = {}^{o}_{f}R^{T}\,{}^{o}\bar{T}$

$^{f}\bar{g} = {}^{o}_{f}R^{T}\,{}^{o}\bar{g}$

Force Equilibrium

$$m\,{}^{f}\bar{g} + {}^{f}\bar{G} + {}^{f}\bar{F} = m\begin{Bmatrix} a_x + V_z\omega_y - V_y\omega_z \\ a_y + V_x\omega_z - V_z\omega_x \\ a_z + V_y\omega_x - V_x\omega_y \end{Bmatrix} = \begin{Bmatrix} A_x \\ A_y \\ A_z \end{Bmatrix}$$

$$^{f}\bar{G} = {}^{o}_{f}R^{T}\,{}^{o}\bar{G} = \bar{A} - {}^{f}\bar{F} - m\,{}^{f}\bar{g}$$

$$\boxed{{}^{o}\bar{G} = {}^{o}_{f}R\,{}^{f}\bar{G}}$$

Moment Equilibrium (about f)

$$^{f}\bar{T} + {}^{f}\bar{c}\times{}^{f}\bar{G} + {}^{f}\bar{M} + {}^{f}\bar{q}\times{}^{f}\bar{F} = \begin{Bmatrix} I_x\alpha_x + (I_z - I_y)\,\omega_y\omega_z \\ I_y\alpha_y + (I_x - I_z)\,\omega_x\omega_z \\ I_z\alpha_z + (I_y - I_x)\,\omega_x\omega_y \end{Bmatrix} = \begin{Bmatrix} B_x \\ B_y \\ B_z \end{Bmatrix}$$

↑ ↑ Find these ...

Note that ${}^{o}c_z = \emptyset$ since foot must touch floor

$$^{o}_{f}R^{T}\left({}^{o}\bar{T} + {}^{o}\bar{c}\times{}^{o}\bar{G}\right) + {}^{f}\bar{M} + {}^{f}\bar{q}\times{}^{f}\bar{F} = \bar{B}$$

$$^{o}\bar{T} + {}^{o}\bar{c}\times{}^{o}\bar{G} = {}^{o}_{f}R\left(\bar{B} - {}^{f}\bar{M} + {}^{f}\bar{q}\times{}^{f}\bar{F}\right) = \bar{D}$$

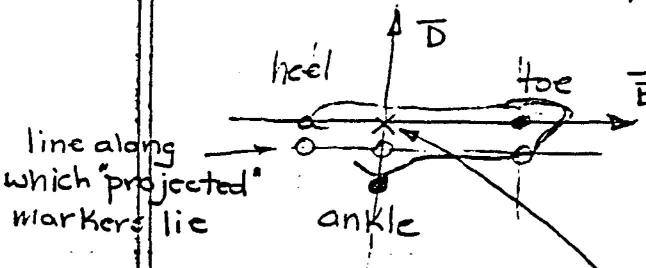
$$\begin{Bmatrix} 0 \\ 0 \\ {}^{\circ}T_z \end{Bmatrix} + \begin{vmatrix} {}^{\circ}C_x & {}^{\circ}C_y & 0 \\ g_x & g_y & g_z \end{vmatrix} = \begin{Bmatrix} D_x \\ D_y \\ D_z \end{Bmatrix}$$

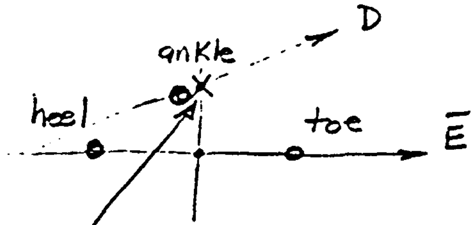
$${}^{\circ}C_y\, g_z = D_x \Rightarrow {}^{\circ}C_y = D_x / g_z$$

$$-{}^{\circ}C_x\, g_z = D_y \Rightarrow {}^{\circ}C_x = -D_y / g_z$$

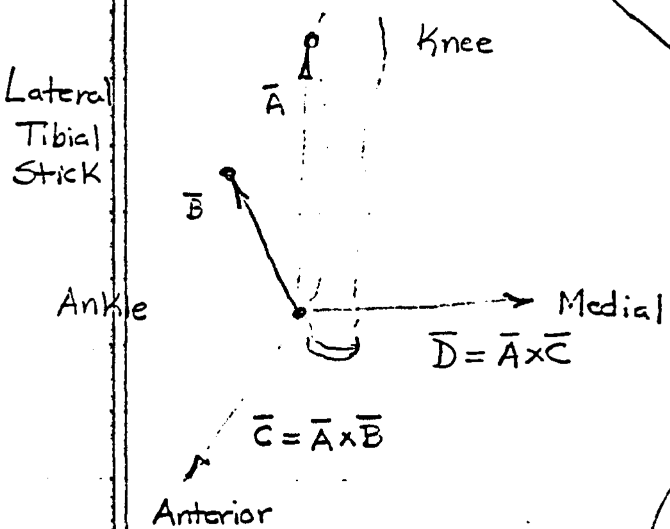
$${}^{\circ}T_z + {}^{\circ}C_x g_y - {}^{\circ}C_y g_x = D_z \Rightarrow {}^{\circ}T_z = D_z + \frac{D_y}{g_z} g_y + \frac{D_x}{g_z} g_x$$

Correction of toe, ankle & heel marker position





Using the ~~standard~~ Boston/OSU bilateral marker set



Compute crossing point x on $\bar{D}$ of line having the minimum distance between $\bar{D}$ & $\bar{E}$.

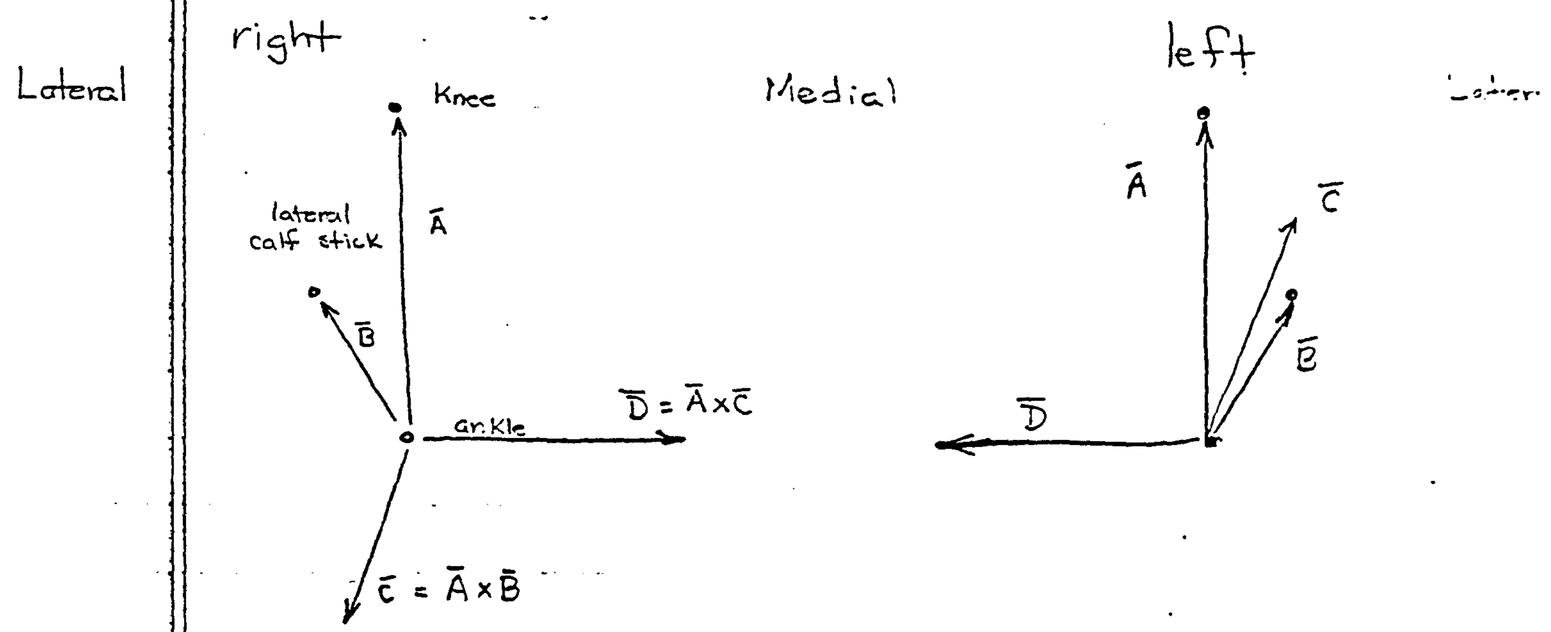
Place ankle marker along $\bar{D}$, midway between original ankle marker & point x. Place heel & toe marker 1/2 that distance laterally along the direction defined by $\bar{D}$.

Not actually used, instead Place ankle marker along $\bar{D}$ at X & leave toe & heel in the same positions.

(See Source Code in AdjustMarkerData routine in Marker.for)

Correction is used to make markers at Knee & ankle more "centered" in the joint.

Ankle &
Knee marker position correction



Based upon my calf dimensions (Dwight)

$$\text{Knee ratio} = \frac{5.25}{42.0}$$

$$\text{Ankle ratio} = \frac{3.5}{42.0}$$

$$\text{Projection distance vector} = \text{ratio} * \frac{\bar{D}}{\|\bar{D}\|}$$

Marker #'s to Segment Correspondence

Rft (1) rt. toe (9) rt. akl (8) rt. heel (20)

lengths: 3
toe-heel
toe-ankle
ankle-heel

Rcf (2) rt. akl (8) rt. tib. stk (24) rt. Knee (7)

lengths: 3
ankle-Knee
ankle-tib. stk
Knee-tib. stk

Rth (3) rt. Knee (7) rt. th. stk (22) rt. hip (4)

lengths: 3
Knee-hip
Knee-th. stk
hip-th. stk

Lft (4) lf. toe (12) lf. akl (11) lf. heel (21)

Lcf (5) lf. akl (11) lf. tib. stk (25) lf. Knee (10)

Lth (6) lf. Knee (10) lf. th. stk (23) lf. hip (5)

Pelvis (7) rt. asis (1) sacral stk (2) lf. asis (3) r hip (4) l hip (5)

lengths: 3
rt. asis - sac. stk
rt. asis - lf. asis
lf. asis - sac. stk

Trunk (8)	rt. asis (1)	lf. asis (3)	rt. sh (14)	lf. sh (17)

lengths: 6
rt asis – lf. asis
rt. asis – rt. sh
rt. asis – lf. sh
lf. asis – lf. sh
lf. asis – rt. sh
rt. sh – lf. sh

R. ua (9)	rt. sh. (14)	rt. el (15)

length 1
rt. sh. – rt. el.

R. la (10)	rt. el. (15)	rt. wr. (16)

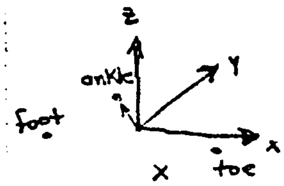
length 1
rt. el. – rt. wr.

L. ua (11)	lf. sh (17)	lf. el (18)
L. la (12)	lf. el (18)	lf. wr (19)

Head ———

Location of segment ends relative to segment LCS

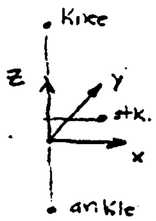
Foot



- force plate position contained in individual foot center of press. data previously calculated in GCS.
- use LCS transform to get force load position in terms of LCS
- " " " " " ankle point in terms of LCS
  - This position is fixed for all frames - use still data

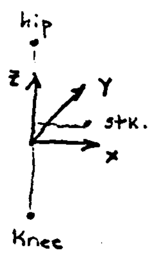
This position changes from frame to frame

Calf



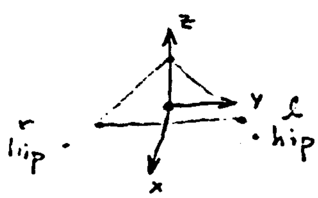
- use LCS transform to get ankle & knee position in terms of LCS
  - average LCS trans. from still data
  - " knee, anke markers " " "

Thigh



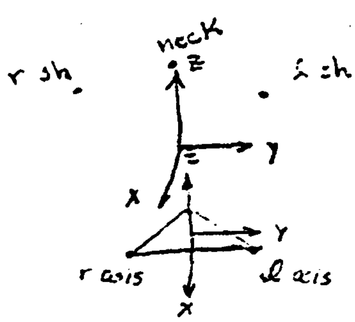
- use LCS transform to get knee & hip position in terms of LCS

Pelvis



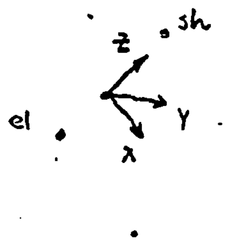
- use LCS transform to get both hips in terms of LCS
- mid section resultant located at LCS center.

Trunk

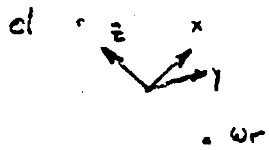


- use LCS transform on r sh, l sh, neck & center of pelvis LCS

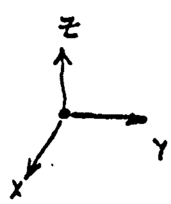
Up arm - locate elbow, shoulder in LCS using LCS transform



Lower Arm - locate elbow, wrist in LCS using LCS transform.



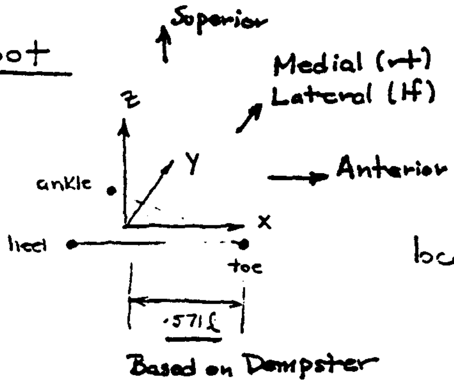
Head - located at LCS center



Location of Segment Coord Systems

Directions marked are for standing position to provide some reference

Foot



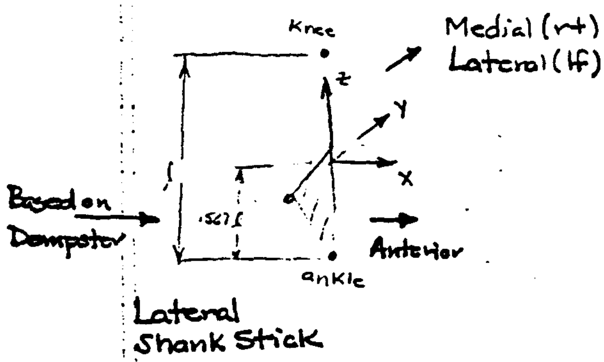
x ∥ to line from heel to toe

y ⊥ to plane formed by heel, ankle & toe

z mutually ⊥ to x & y

located at geometric center Δ

Calf

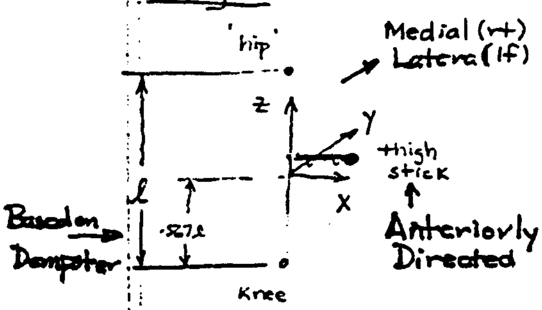


z from ankle to knee

x ⊥ z & ankle-shank-knee plane

y mutually ⊥ to x & z

Thigh

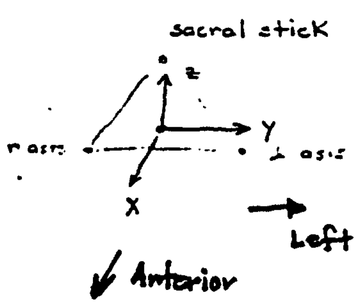


z from knee to hip

y ⊥ z & knee-thigh-hip plane

x mutually ⊥ to y & z

Pelvis



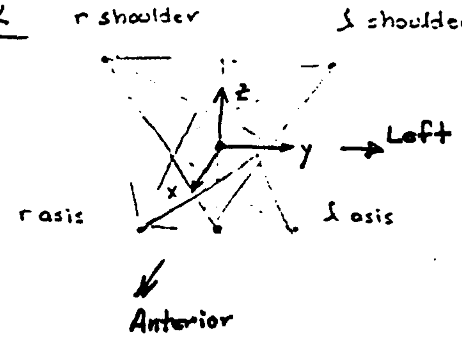
located at the geometric center Δ formed by r asis, sacral stick, & l asis

y from r asis to l asis

z ⊥ to plane of Δ

x mutually ⊥ to y & z

Trunk



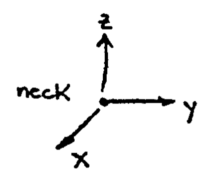
located at [the average of the] geometric center of Δ's formed by r, l shoulder & average of r, l asis & r, l asis & average of r, l shoulder

y: ∥ to line from r to l shoulder

x: ⊥ plane of r, l shoulder & average of r, l asis

z: mutually ⊥ to x & y

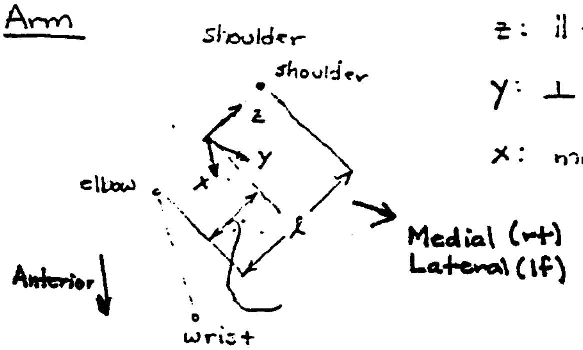
Head



located at neck marker

∥ to lab coord system

Upper Arm

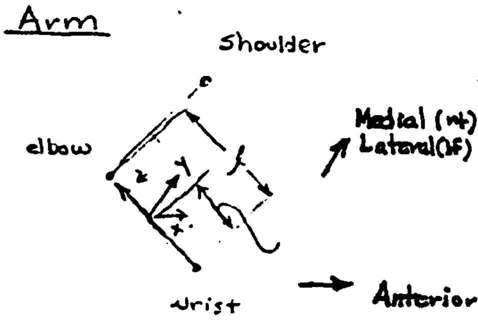


z: ∥ to line from elbow to shoulder

y: ⊥ to plane of shoulder, elbow, wrist

x: mutually ⊥ to x & z

Lower Arm



z: ∥ to line from wrist to elbow

y: ⊥ to plane of shoulder, elbow, wrist

x: mutually ⊥ to x & z

Numerical Integration of experimental Data

For numerical integration of experimentally acquired data, there are two different types required.

1) Moving integration window of selectable width
   - Used with EMG processing
   - Use extended trapezoidal rule (Num. Recipes, p. 107, 111)

$$\int_{t-\tau/2}^{t+\tau/2} f(x)\,dx = h\left[\tfrac{1}{2} f_{i-\frac{N}{2}} + f_{i+1-\frac{N}{2}} + f_{i+2-\frac{N}{2}} + \dots + f_{i-1+\frac{N}{2}} + \tfrac{1}{2} f_{i+\frac{N}{2}}\right]$$

where t = center point or time at which integration takes place.
Sample i corresponds to t.
h is the sample interval (sec)
N is the number of samples in the window
τ is the window width (sec)

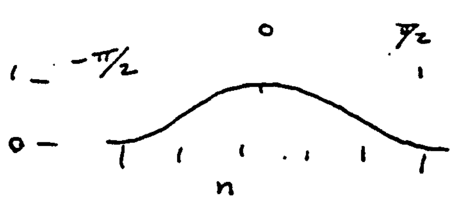
$$N = \text{Int}\left(\frac{\tau}{h}\right)$$

1st pt   last pt

$$f'_i = \sum_{j=i-\frac{N}{2}+1}^{i+\frac{N}{2}-1} f_j + \tfrac{1}{2}\left(f_{i-\frac{N}{2}} + f_{i+\frac{N}{2}}\right)$$

for any i where $i+1-\frac{N}{2} < j < i$ is less than 1, set $f'_i = 0$

for any i where $i < j < i-1+\frac{N}{2}$ is greater than N, set $f'_i = 0$



$$\cos\left(-\frac{\pi}{2} + \pi\,\frac{i-1}{n-1}\right)$$

2) Standard Summation integration which continually adds the previous intervals

$f_1' = 0$

$f_2' = \frac{h}{2}[f_1 + f_2]$

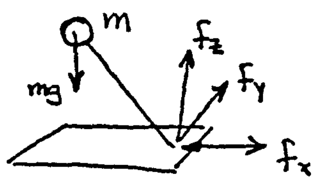
$f_3' = h\left[\frac{1}{2}f_1 + f_2 + \frac{1}{2}f_3\right]$

$f_4' = h\left[\frac{1}{2}f_1 + f_2 + f_3 + \frac{1}{2}f_4\right]$

$\Rightarrow \quad f_i' = f_{i-1}' + \frac{h}{2}[f_{i-1} + f_i]$ Nice & Simple...

Determination of Center of Mass of Body based upon force plate data...

Assume body is a single lumped mass attached to a massless stick striking the force plate



Since the force plate reference frame is stationary (as opposed for instance to the foot LCS which is used for calculating the ankle loads), Then,

$$+m\bar{g}+\bar{F}=m\bar{a}$$

Note that $\bar{F}$ is the force of the plate upon the body whereas usually the body on the plate is what is measured.

$$\begin{Bmatrix} f_x \\ f_y \\ f_z \end{Bmatrix} = m \begin{Bmatrix} a_x \\ a_y \\ a_z - g \end{Bmatrix}, \quad \bar{F}(t) \ \& \ \bar{a}(t)$$

$$\bar{a} = \ddot{p} = \frac{\bar{F}}{m} + \bar{g}$$

$$\ddot{x} = \frac{1}{m} f_x$$

$$\dot{x} = \frac{1}{m} \int f_x \, dt + V_{x_0}$$

$$x = \frac{1}{m} \int_0^T \left(\int_0^T f_x \, dt \right) dt + V_{x_0} T + x_0$$

Ant/Post position of force

$$\ddot{y} = \frac{1}{m} f_y$$

$$\dot{y} = \frac{1}{m} \int_0^T f_y \, dt + V_{y_0}$$

$$y = \frac{1}{m} \int_0^T \left(\int_0^T f_y \, dt \right) dt + V_{y_0} T + y_0$$

med/lat position of force

$$\ddot{z} = \frac{1}{m}(f_z - g)$$

$$\dot{z} = \frac{1}{m}\int_0^T f_z\,dt - \frac{gT}{m} + V_{z_0}$$

$$z = \frac{1}{m}\int_0^T\left(\int_0^T f_z\,dt\right)dt - \frac{1}{2}\frac{g}{m}T^2 + V_{z_0}T + z_0$$

Now if x_0, y_0, z_0 & $V_{x_0}, V_{y_0}, V_{z_0}$ are not known at $t=0$ but rather, at $t = t_1$, then ?....

$$x = \frac{1}{m}\int_0^T\left(\int_0^T f_x\,dt\right)dt + V_{x_0}(T - t_1) + x_0$$

$$y = \frac{1}{m}\int_0^T\left(\int_0^T f_y\,dt\right)dt + V_{y_0}(T - t_1) + y_0$$

where $x = x(T)$, $T = 0, h, 2h,$
h = time increment

$$\dot{z} = \frac{1}{m}\int_0^T f_z\,dt - \frac{g}{m}(T - t_1) + V_{z_0}$$

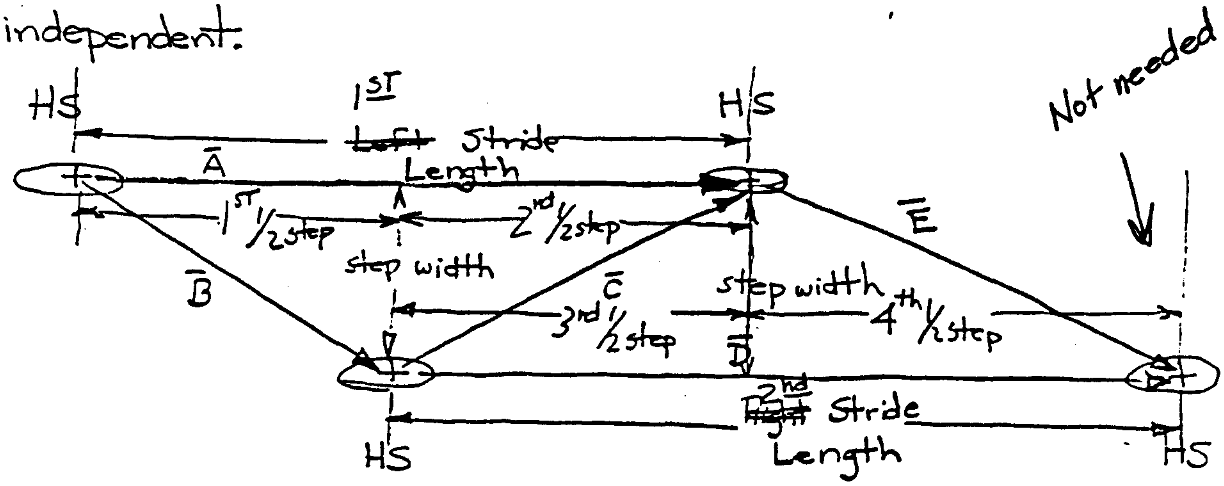
$$z = \frac{1}{m}\int_0^T\left(\int_0^T f_z\,dt\right)dt - \frac{1}{2}\frac{g}{m}(T - t_1)^2 + V_{z_0}(T - t_1) + z_0$$

Integrals must be evaluated for each time step to get the CM position at all time steps.

See Notes on integration

Stride Parameters

- Center of foot (i.e. foot local coordinate system center) is used in calculations. This makes calculations marker set independent.



If information is available on successive pair of heel strikes from both sides, then $\bar{D}$ & $\bar{E}$ are calculated.

1^{ST} stride length $= \|\bar{A}\|$

2^{nd} stride length $= \|\bar{D}\|$

Side_Stride_Ratio = Stride_Length(1^{ST}) / Stride_Length(2^{nd})

$N_A = \frac{\bar{A}}{\|\bar{A}\|}$, $N_D = \frac{\bar{D}}{\|\bar{D}\|}$

1^{st} stride:

1^{ST} HS to 2^{nd} HS step width $= \sqrt{\|\bar{B}\|^2 - F^2}$ ...

1^{ST} HS to 2^{nd} HS step length $= \bar{B} \cdot \bar{N}_A = F$

2^{nd} HS to 3^{rd} HS step width $= \sqrt{\|\bar{C}\|^2 - G^2}$

2^{nd} HS to 3^{rd} HS step length $= \bar{C} \cdot \bar{N}_A = G$

These will be the same size since they are the same ⊥ bisector of ΔABC.

2^{nd} stride (Optional):

2^{nd} HS to 3^{rd} HS step length $= \bar{C} \cdot \bar{N}_D = H$

2^{nd} HS to 3^{rd} HS step width $= \sqrt{\|\bar{C}\|^2 - H^2}$

3^{rd} HS to 4^{th} HS step length $= \bar{E} \cdot \bar{N}_D = I$

3^{rd} HS to 4^{th} HS step width $= \sqrt{\|\bar{E}\|^2 - I^2}$

(Opt) Side_width_ratio = Step_width (1^{ST}) / Step_width(2^{nd})

(Opt) Step_width_avg =

These are the same length. (same reason as above)

(sec) Cycle_Time (1^{ST}) = (Frame # 1^{ST} HS − Frame # 3^{rd} HS) * Camera_Speed (1/...)

(Opt) Cycle_Time(2^{nd}) = (Frame # 2^{nd} HS − Frame # 4^{th} HS) * Camera_Speed

(Opt) Side_Time_Ratio = Cycle_Time (1^{ST}) / Cycle_Time (2^{nd})

Cadence = (1.0/ Cycle_Time(1st)) * 2 * 60 sec/min (step/min)
or

(opt) Cadence = 1.0/(Cycle_Time(1st) + Cycle_Time(2nd)) * 60

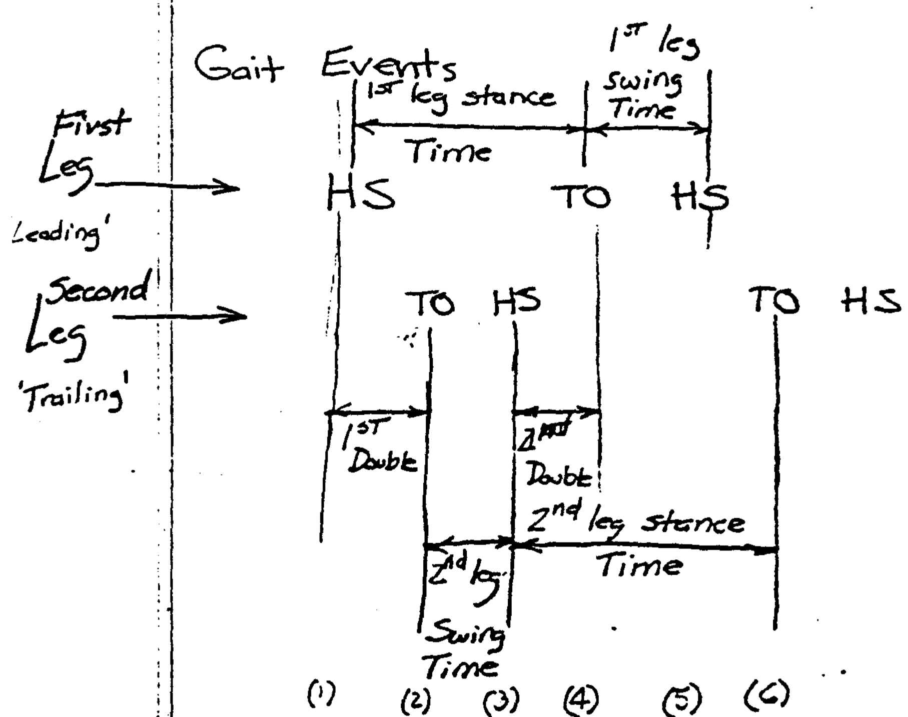
Velocity(1st) = Stride_Length(1st) / Cycle_Time(1st)

(opt) Velocity(2nd) = Stride_Length(2nd) / Cycle_Time(2nd)

(opt) Velocity_Avg = (Velocity(1st) + Velocity(2nd)) / 2

(opt) Side_Velocity_Ratio = Velocity(1st) / Velocity(2nd)

Gait Events



Stance_Time(1st) = (4) - (1)

Swing_Time(1st) = (5) - (4)

(optional) Stance_Time(2nd) = (6) - (3)

Swing_Time(2nd) = (3) - (2)

Double_Time(1st) = (2) - (1)

Double_Time(2nd) = (4) - (3)

To make % of cycle, divde these quantities by Cycle_Time_Avg

NOTE: Additional Equations are included in softwa for women and children (references in code comments)

Position of CM based upon segment length

Dempster (1955)

	length ratio			
foot	.571 from distal joint		Up arm	.564
shank	.567		Lw arm	.570
thigh	.567			

Segment mass - use predictive equations of McConville (1980) along with density data of Dempster (1955)

foot $m = 1.10 * d_w * (^{9.31} * Ht + ^{1.87} * Wt - ^{994})$

shank $m = 1.09 * d_w * (26.72 * Ht + 11.84 * Wt - 2912)$

thigh $m = 1.05 * d_w * (35.19 * Ht + 45.30 * Wt - 4083)$

d_w = density of water, Kg/cm³

Ht = total hieght, cm

Wt = body weight, lbs

m = mass, Kg

Segment inertia - use predictive equations of McConville (1980) (principal moments of inertia)

foot a-p $I_x = 110 * Ht + 30 * Wt - 16550$

m-l $I_y = 948 * Ht + 71 * Wt - 137353$

$I_z = 954 * Ht + 86 * Wt - 138369$

shank a-p $I_x = 12323 * Ht + 1702 * Wt - 1907428$

m-l $I_y = 12429 * Ht + 1730 * Wt - 1922497$

$I_z = 291 * Ht + 477 * Wt - 66811$

thigh a-p $I_x = 28839 * Ht + 6407 * Wt - 4659953$

m-l $I_y = 28559 * Ht + 7298 * Wt - 4679995$

$I_z = -1587 * Ht + 4537 * Wt - 72496$

$gm \cdot cm^2 \frac{Kg}{1000 gm} \frac{m^2}{10000 cm^2}$

I = principal moment of inertia, $gm \cdot cm^2$

Segment Mass - Trunk & Arms

Pelvis $m = 1.01 * dw * (-68.97 * Ht + 98.98 \times wt + 6765)$ (p.41)

Thorax/Abdomen $m = 0.92 * dw * (-22.71 * Ht + 163.68 * wt + 524)$ (p.37) (1.09783)

$+ 1.01 * dw * (-23.05 * Ht + 19.42 * wt + 3121)$ (p.39)

(-25.30) (21.32) (2426

Head/neck $m = 1.0[illegible] * dw * (6.84 * Ht + 2.05 * wt + 2806)$ (p.33)

$+ 1.11 * dw * (-5.50 * Ht + 6.32 * wt + 943)$ (p.35)

Up arm $m = 1.07 * dw * (-2.14 * Ht + 13.25 * wt + 76)$ (p.43)

Lw arm $m = 1.13 * dw * (-.36 * Ht + 10.01 * wt + 267)$ (p.71)

Segment Moment of inertia

Pelvis $I_Y = -10851 * Ht + 13750 * wt + 492711$

$I_x = -10283 * Ht + 14215 * wt + 396174$ (p.41)

$I_z = -14684 * Ht + 17498 * wt + 772{,}875$

Thorax $I_y' = 3738 * Ht + 36636 * wt - 4768961$

$I_x = 8183 * Ht + 47484 * wt - 3393924$ (p.37)

$I_z = -14325 * Ht + 36254 * wt - 675725$

Head $I_y = 1097 * Ht + 103 * wt + 36972$

$I_x = 859 * Ht + 86 * wt + 20735$ (p.33)

$I_z = 216 * Ht + 168 * wt + 83847$

Up arm $I_y = 934 * Ht + 1094 * wt - 224626$

$I_x = 627 * Ht + 1004 * wt - 198020$ (p.43)

$I_z = -338 * Ht + 391 * wt + 19102$

Lw arm $I_y = 4086 * Ht + 1316 * wt - 645877$

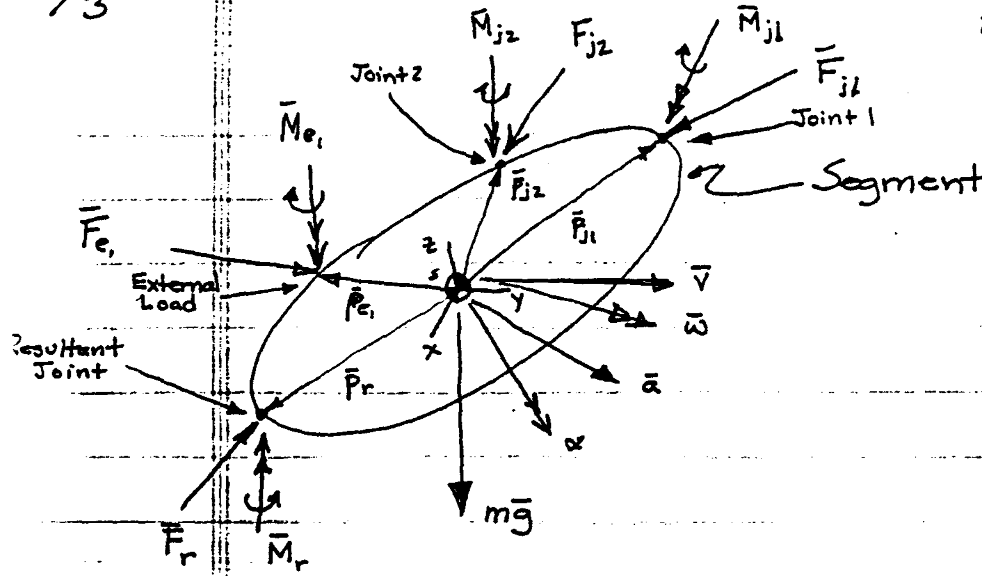
$I_x = 4058 * Ht + 1316 * wt - 642117$ (p.71)

$I_z = -139 * Ht + 200 * wt + 7973$

Description of Net Joint Resultant load calculations

ANZ carries sequential calculations from segment to segment the evaluate the net resultant joint loads. The actual sequence may vary depending upon what GRL data is known but the calculation of the joint loads for an individual segment is the same in all case — with one minor correction. A number of sequences are described in other descriptions of body/joint connectivity. The correction mentioned is whether the calculation of the net joint load is done in a sequence progressing from distal-to-proximal or proximal-to-distal. The normal sequence is distal-to-proximal so whenever the other order occurs the sign of the joint resultant load (JRL) must be negated. In the program, the JRL's are calculated relative to the segment local coordinate System (LCS) in terms of proximal segment acting upon distal segment. Following this, the JRL is transformed into 3 other forms relative to the proximal segment LCS, the global CS and the non orthogonal Joint CS derived from the euler angle sequence (Flexion/Ext. – Ab/Adduction – In/External).

The following segment illustrates the basic situation being evaluated along with the equations used in this evaluation:



All loads, positions, velocities, & accelerations are relative to segment LCS!

$\bar{F}_{ji}, \bar{M}_{ji}$ - i^{th} joint load

$\bar{F}_{ei}, \bar{M}_{ei}$ - i^{th} external load

$\bar{F}_r, \bar{M}_r$ - Resultant Joint load

$\bar{p}_{ji}$ - location of joint load i

$\bar{p}_{ei}$ - " " external load i

$\bar{p}_r$ - " " resultant joint l[oad]

$\bar{g}$ - gravity vector

There may be multiple joints affecting the segment whose net JRL ($\bar{F}_j$ - force, $\bar{M}_j$ - moment) have already been computed in a previous evaluation of the sequence, n_j - number of joints. Also, there maybe multiple external loads (n_e) applied to the segment although in practice usually only a single GRL (ground reaction load) is applied to the foot segment only. The locations of all joints & external loads are known through either definition or calculation. (An Assumption)

$\bar{v}$ - translational velocity of segment

$\bar{\omega}$ - rotational " "

$\bar{a}$ - translational acceleration "

$\bar{\alpha}$ - rotational " "

m - mass

I_x, I_y, I_z - Prinicipal Inertias

Another Assumption - Segment LCS & principal inertial axes are aligned with one another & LCS is located at the center of mass. This is estimated using anthropometric equations (see other notes).

Since the dynamics equations are evaluated relative to the segment LCS which is rotating & translating, the following equations apply: (about the segment LCS)

Σ Forces : $\bar{E} = m\bar{g} + \sum_{i=1}^{n_j} \bar{F}_{ji} + \sum_{i=1}^{n_e} \bar{F}_{ei} + \bar{F}_r$

Σ Moments : $\bar{J} = \sum_{i=1}^{n_j} \bar{M}_{ji} + \sum_{i=1}^{n_j} \bar{p}_{ji} \times \bar{F}_{ji} + \sum_{i=1}^{n_e} \bar{M}_{ei} + \sum_{i=1}^{n_e} \bar{p}_{ei} \times \bar{F}_{ei} + \bar{p}_r \times \bar{F}_r$

where : $\bar{E} = m \begin{Bmatrix} a_x + v_z\omega_y - v_y\omega_z \\ a_y + v_x\omega_z - v_z\omega_x \\ a_z + v_y\omega_x - v_x\omega_y \end{Bmatrix}$

$$\bar{J} = \begin{Bmatrix} I_x\alpha_x + (I_z - I_y)\,\omega_y\omega_z \\ I_y\alpha_y + (I_x - I_z)\,\omega_x\omega_z \\ I_z\alpha_z + (I_y - I_x)\,\omega_x\omega_y \end{Bmatrix}$$

The derivation of quantities $\bar{E}$ & $\bar{J}$ can be found in most any Advanced Dynamics textbook. For example,

Advanced Dynamics: Modeling & Analysis by A. Frank D'Souza & Vijay K. Garg Prentice-Hall, 1984. (see p. 96-97, in particular Eq. 4.44 & 4.47)

To find the resultant joint loads, first compute $\bar{F}_r$

$$\bar{F}_r = \bar{E} - m\bar{g} \quad \sum_{i=1}^{n_j} F_{ji} \quad \sum_{i=1}^{n_e} F_{ei}$$

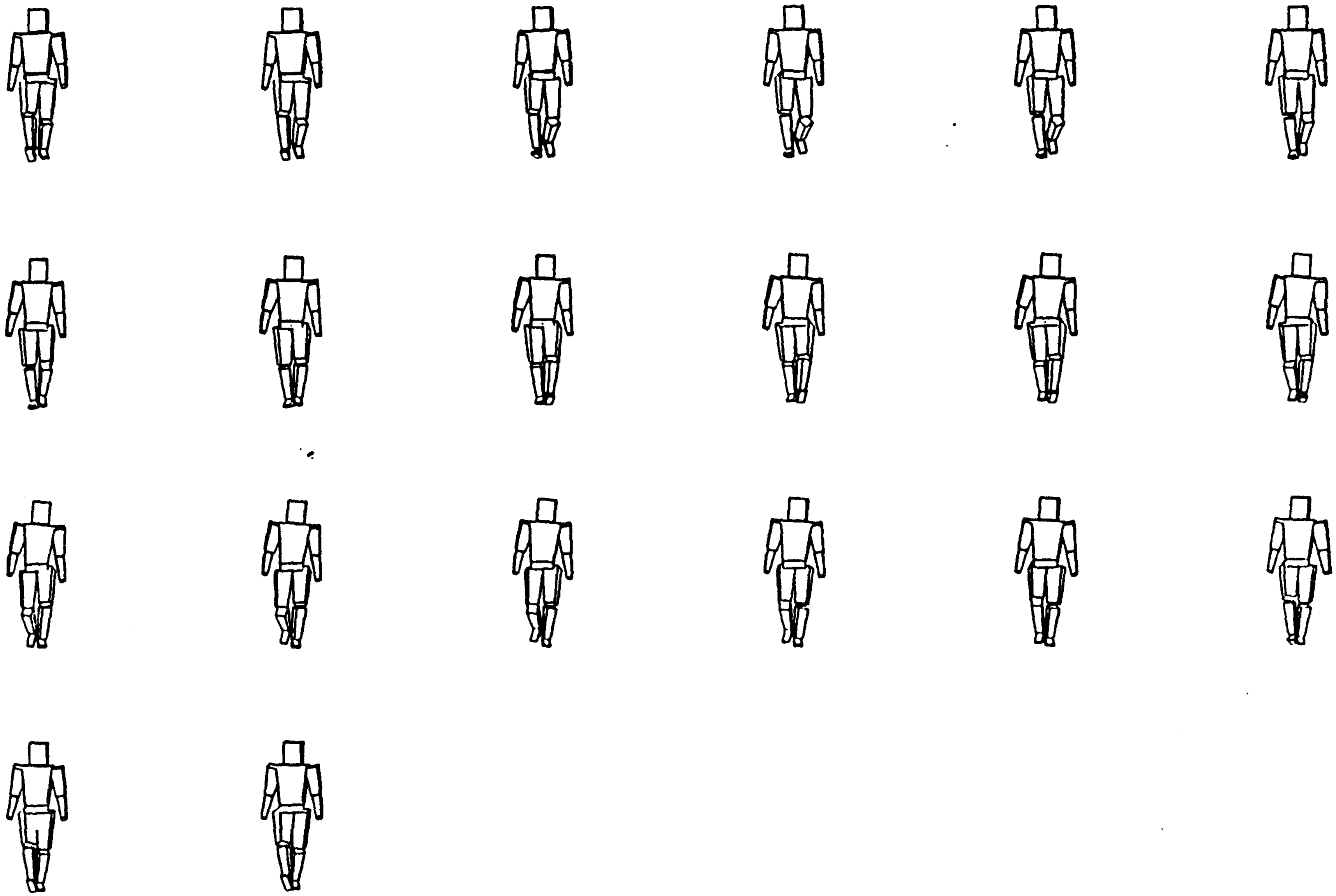
Then solve for $\bar{M}_r$ now that $\bar{F}_r$ is known

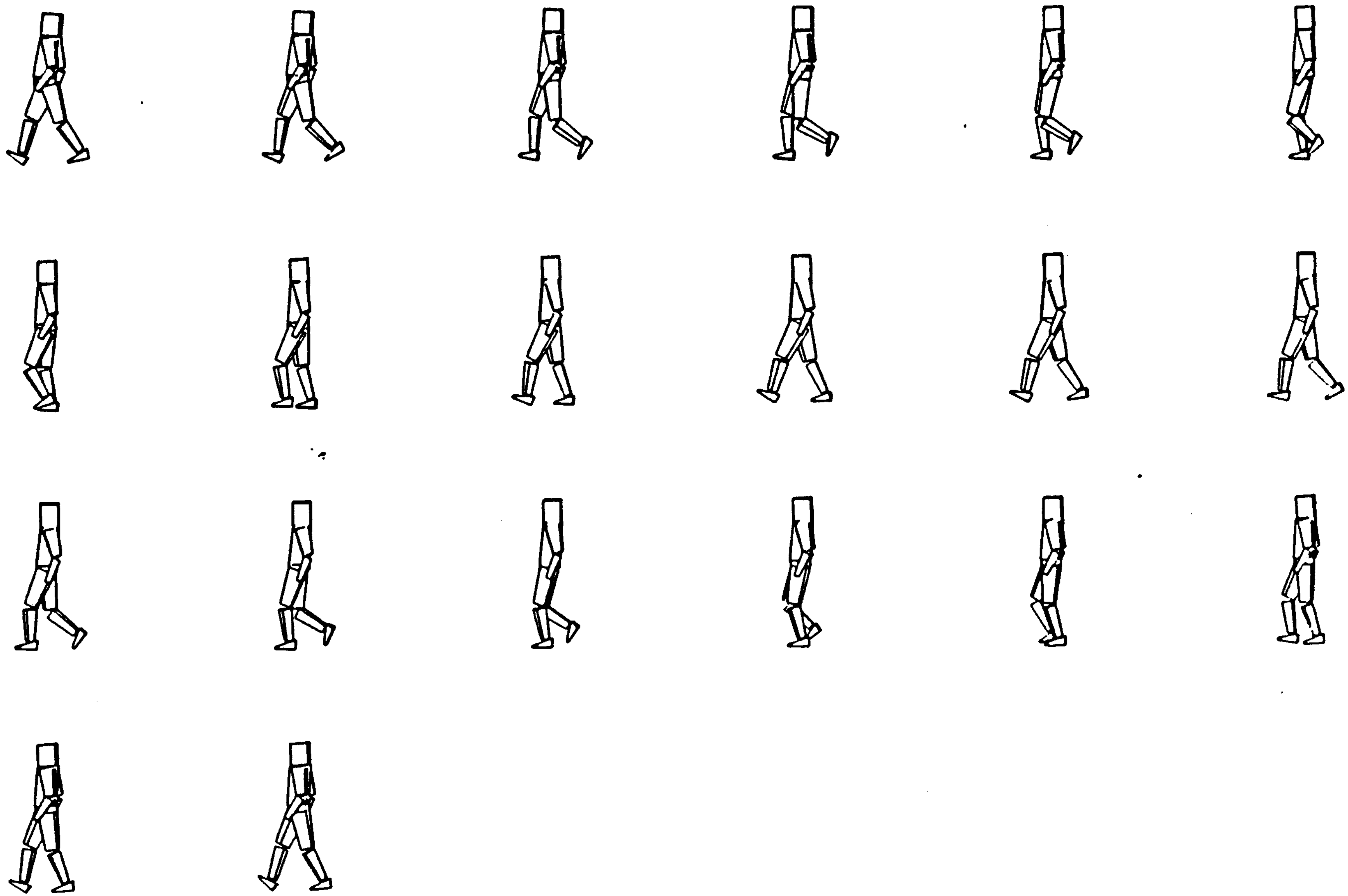
$$\bar{M}_r = \bar{J} - \sum_{i=1}^{n_j} M_{ji} - \sum_{i=1}^{n_j} \bar{p}_{ji} \times \bar{F}_{ji} - \sum_{i=1}^{n_e} M_{ei} - \sum_{i=1}^{n_e} \bar{p}_{ei} \times \bar{F}_{ei} - \bar{p}_r \times \bar{F}_r$$

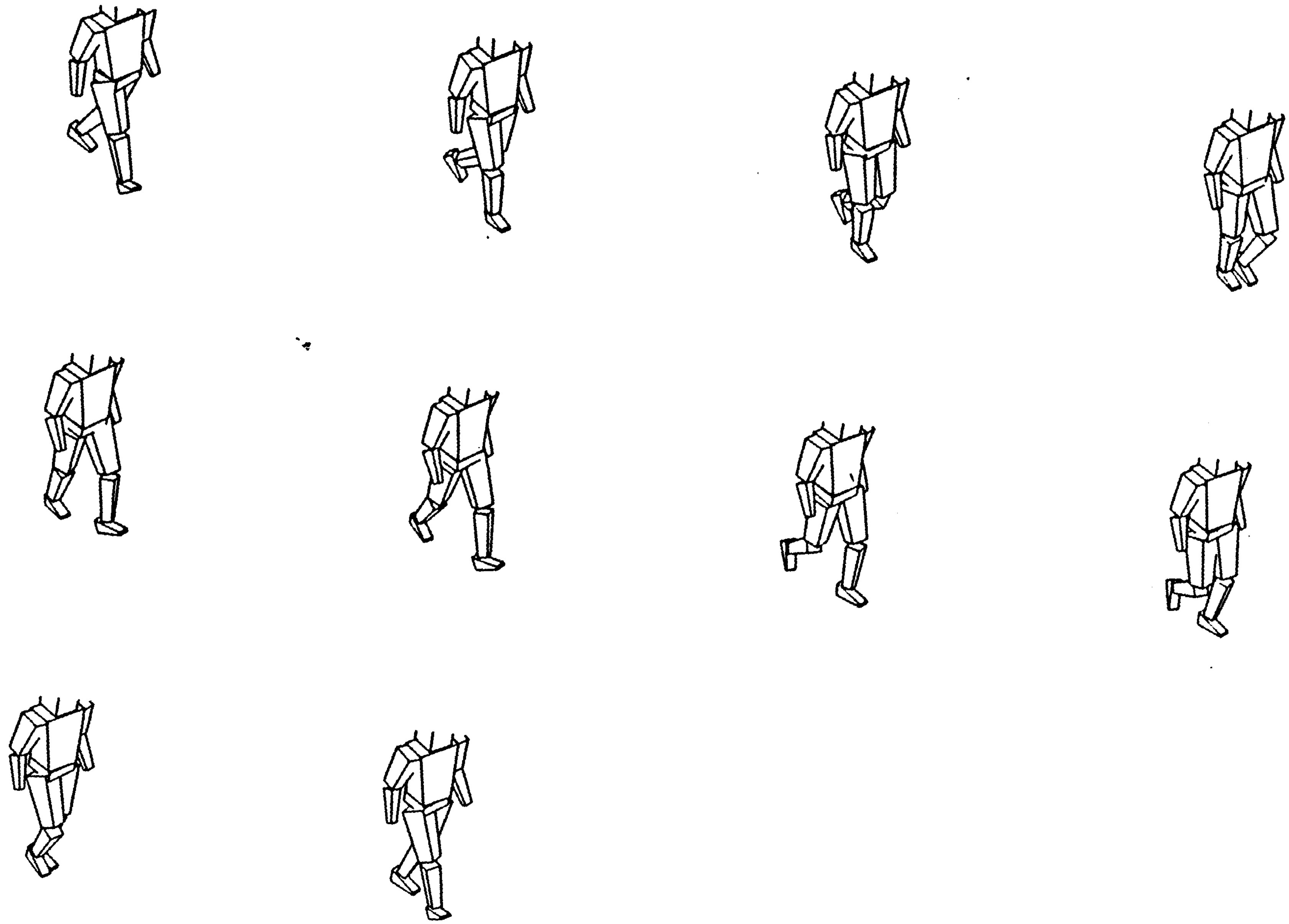
These equations can be found in the routine CalcAJntLoad.

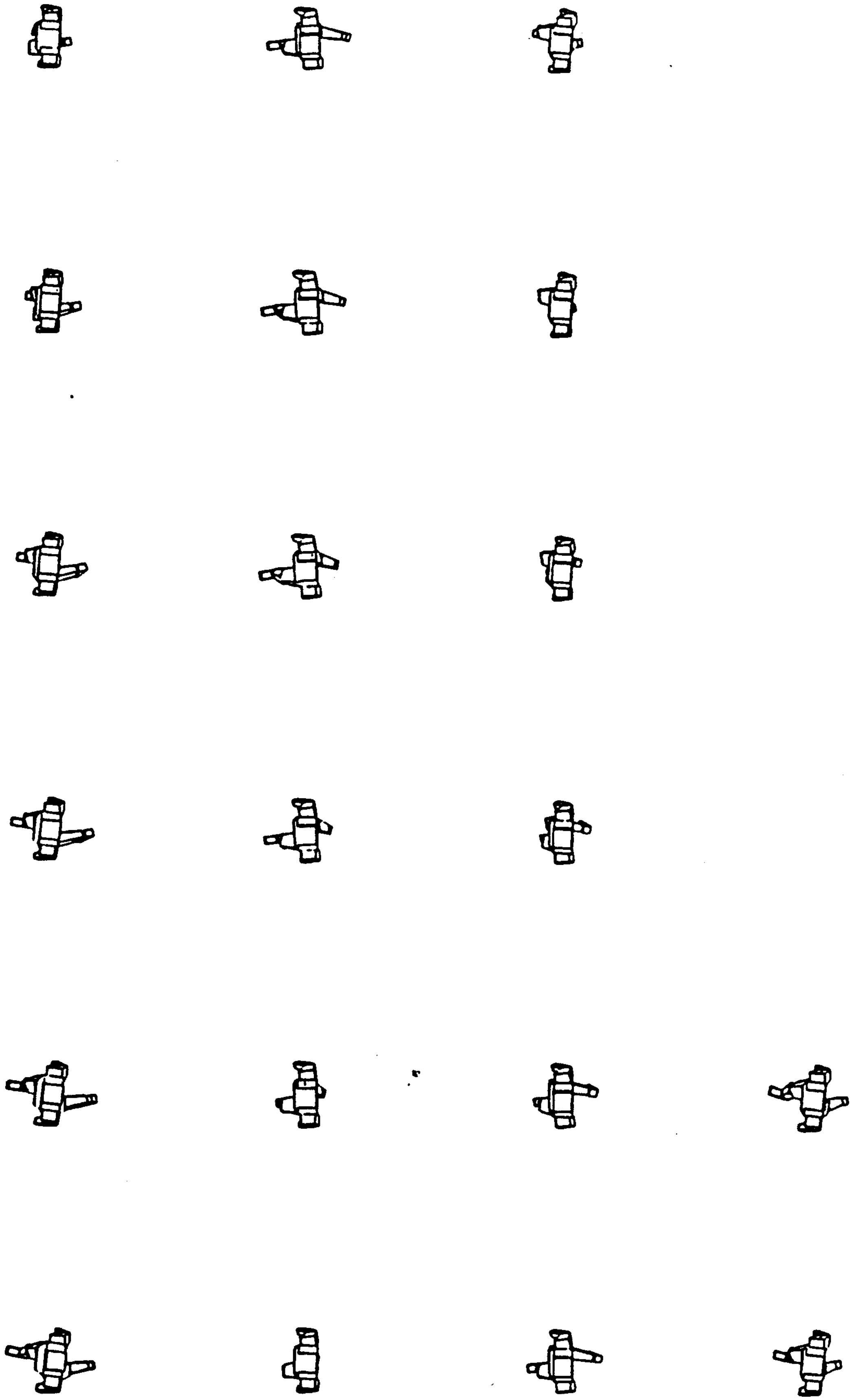
The full sequence of calculations is found in CalcAllJntLoads

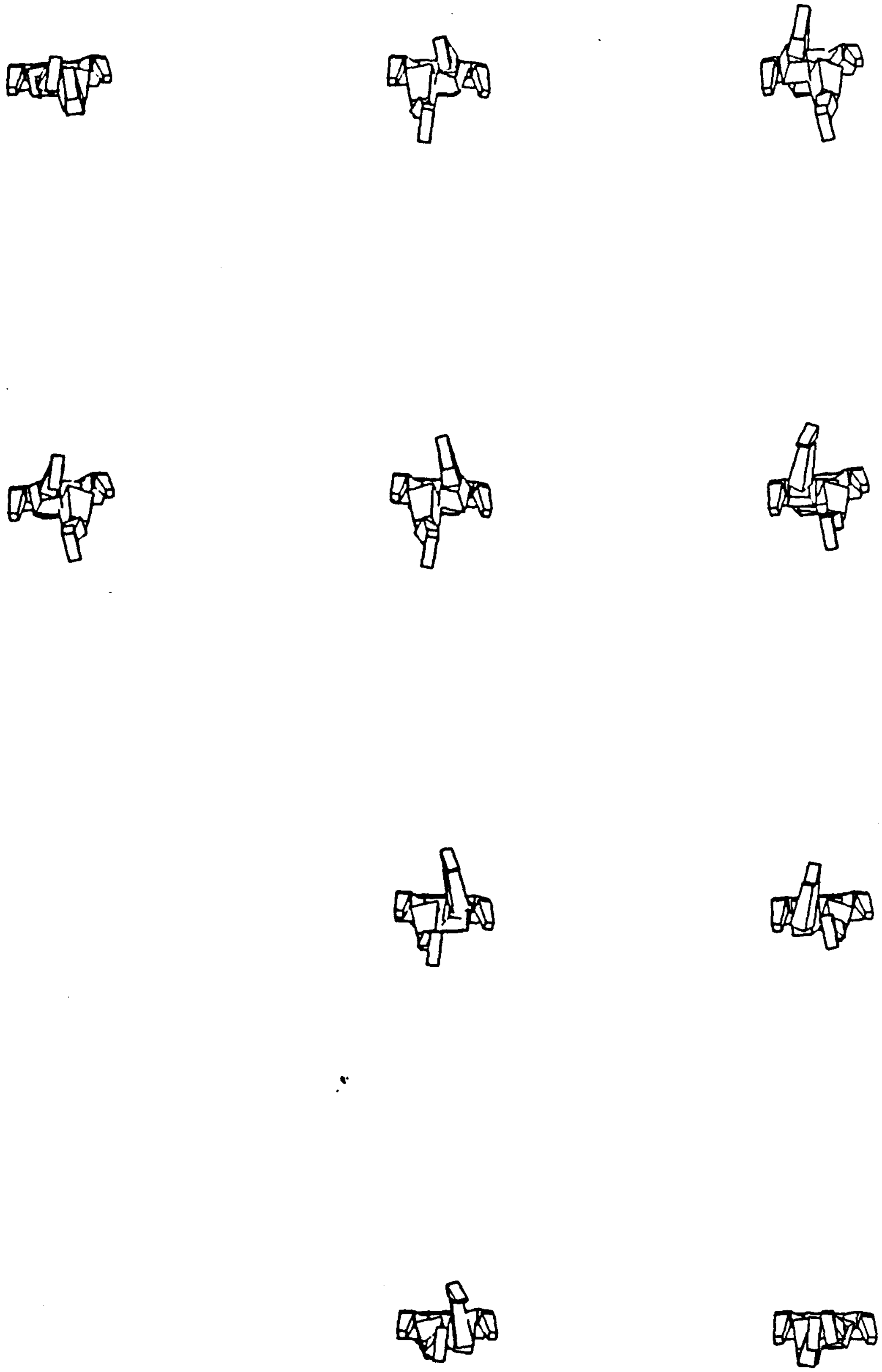
Sample graphics generated in TELIO using data calculated with ANALYZE



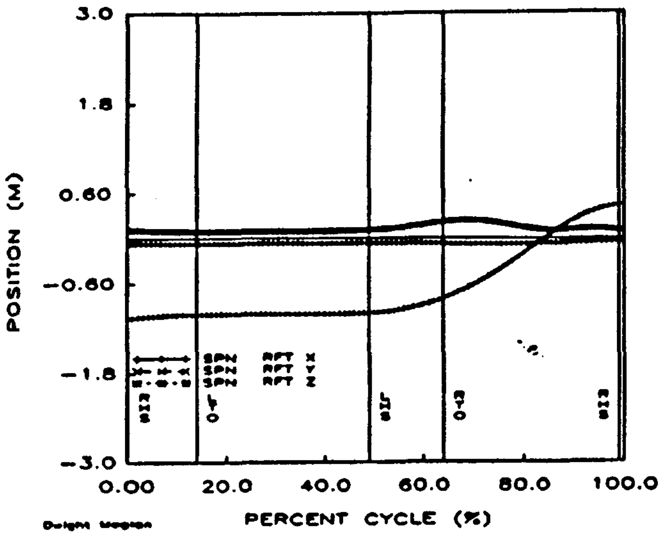




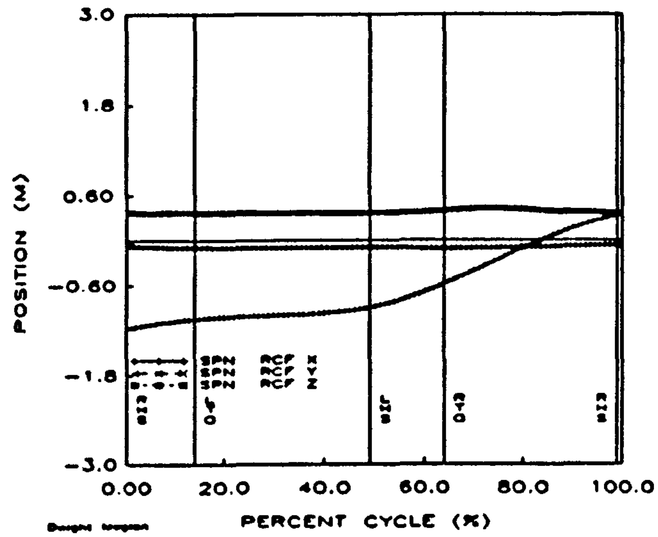




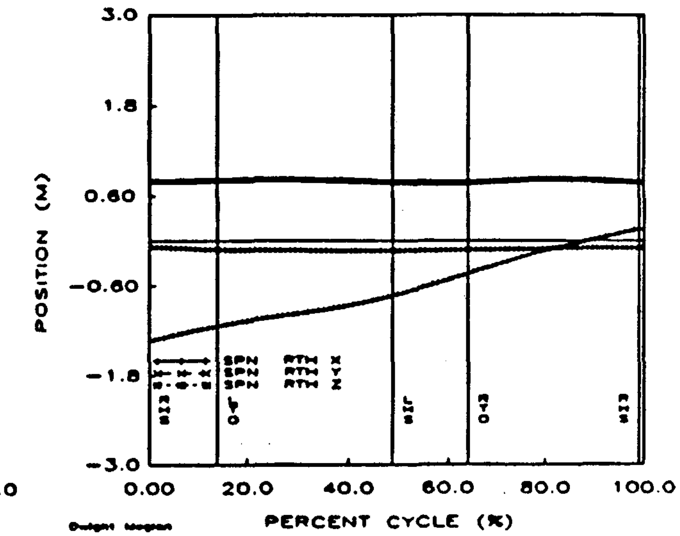
SEGMENT POSITION
RIGHT FOOT



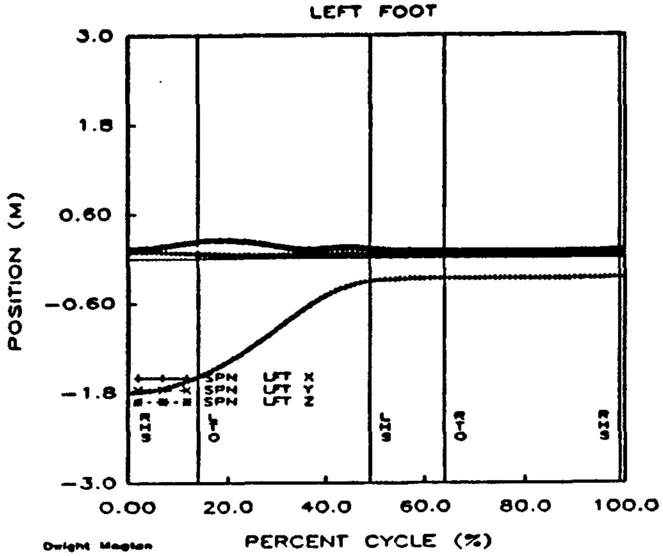
SEGMENT POSITION
RIGHT CALF



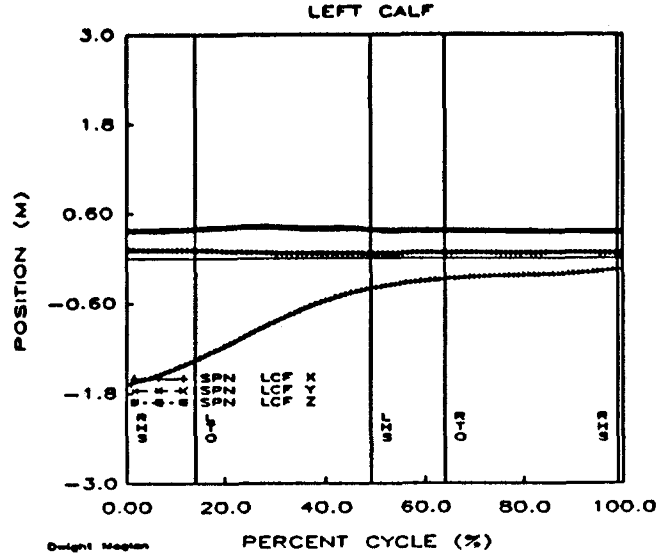
SEGMENT POSITION
RIGHT THIGH



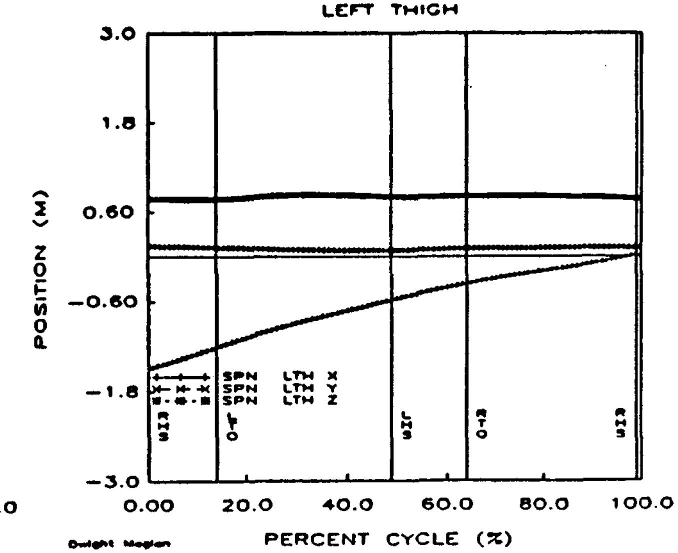
SEGMENT POSITION
LEFT FOOT



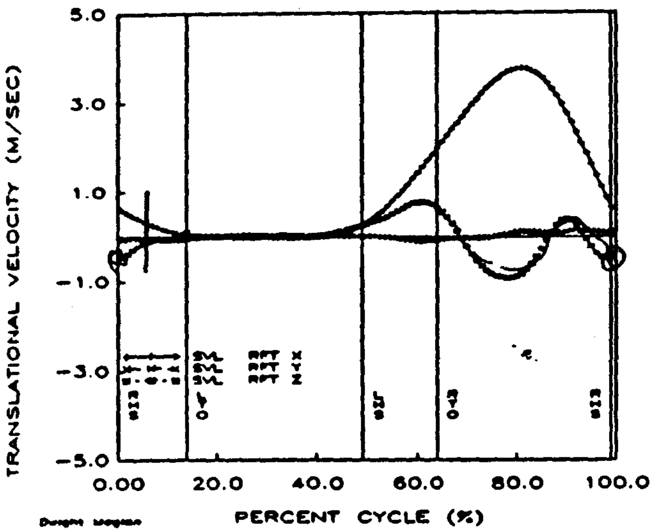
SEGMENT POSITION
LEFT CALF



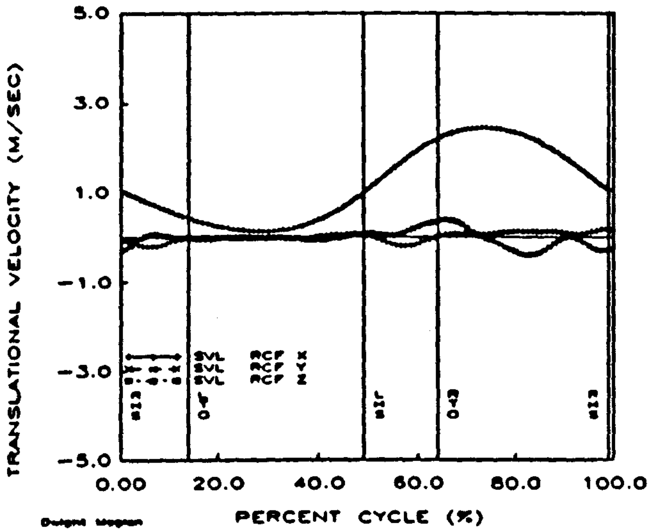
SEGMENT POSITION
LEFT THIGH



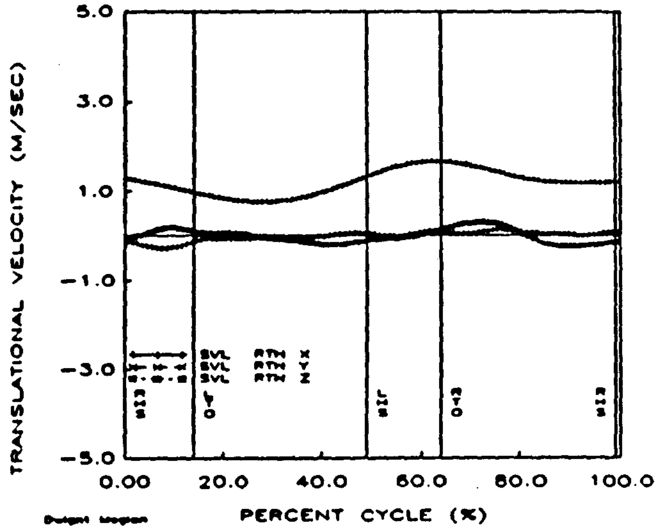
SEGMENT TRANSLATIONAL VELOCITY
RIGHT FOOT



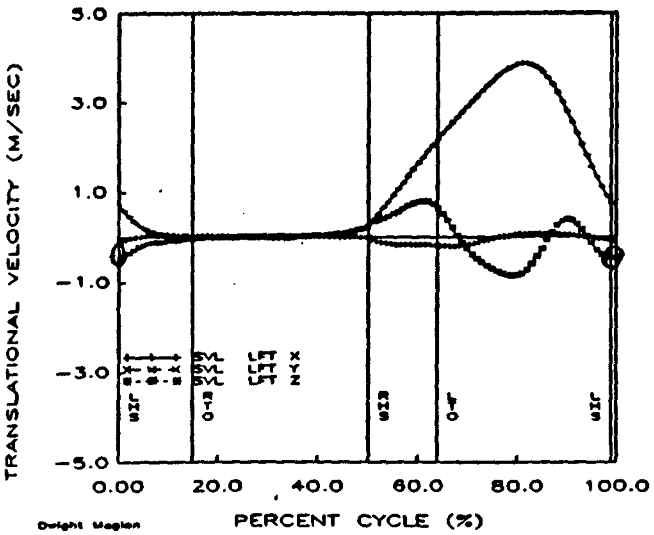
SEGMENT TRANSLATIONAL VELOCITY
RIGHT CALF



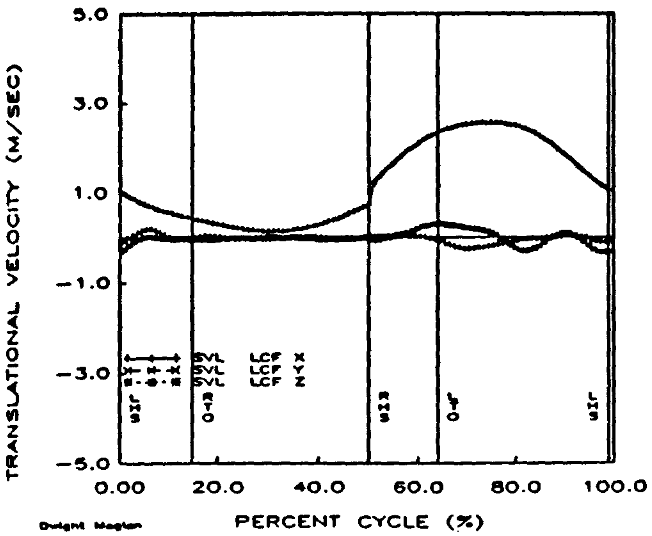
SEGMENT TRANSLATIONAL VELOCITY
RIGHT THIGH



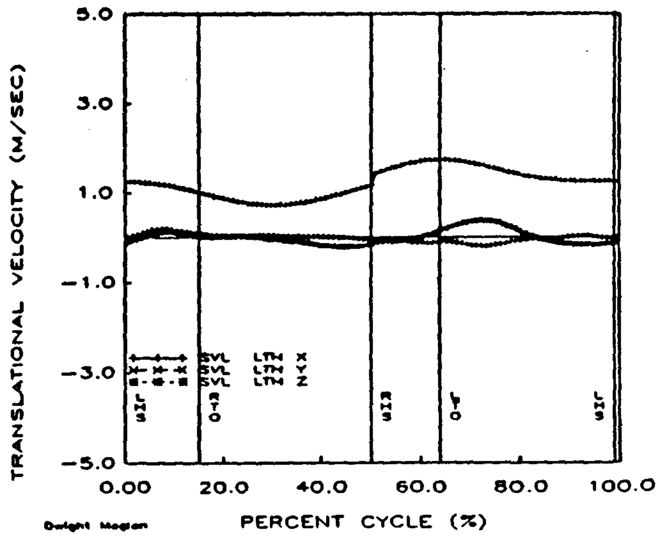
SEGMENT TRANSLATIONAL VELOCITY
LEFT FOOT

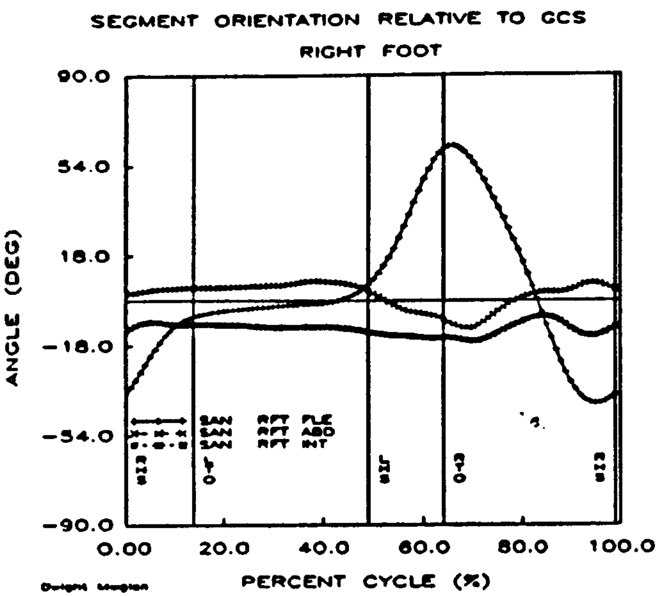


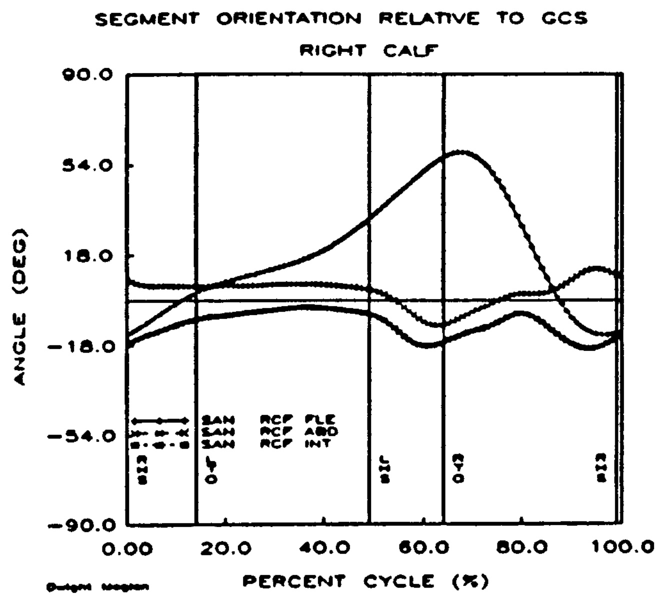
SEGMENT TRANSLATIONAL VELOCITY
LEFT CALF

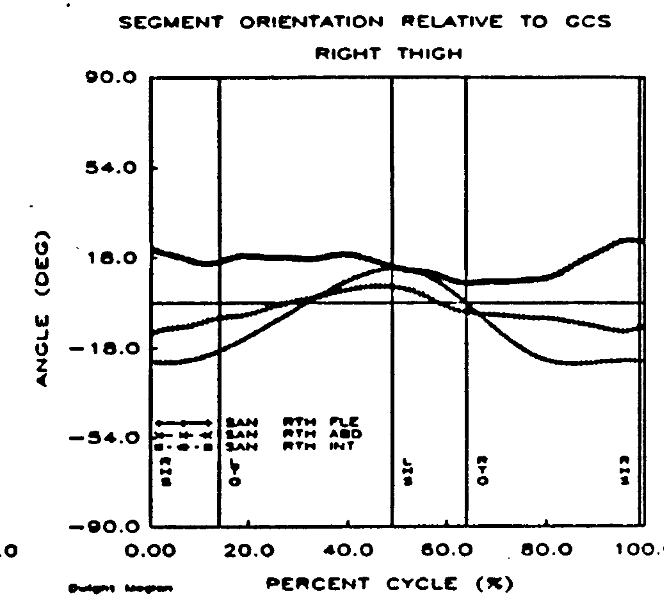


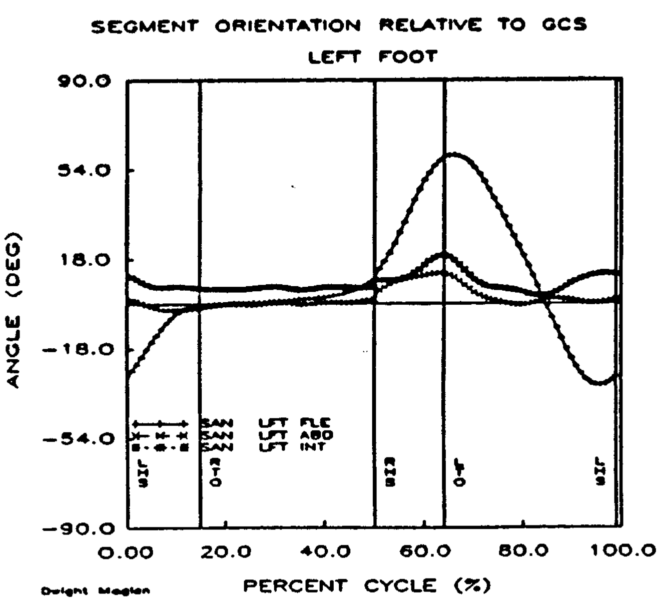
SEGMENT TRANSLATIONAL VELOCITY
LEFT THIGH

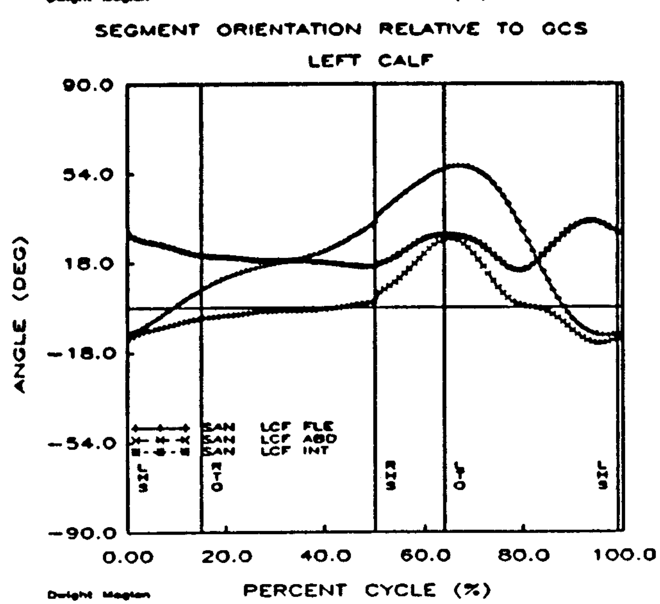


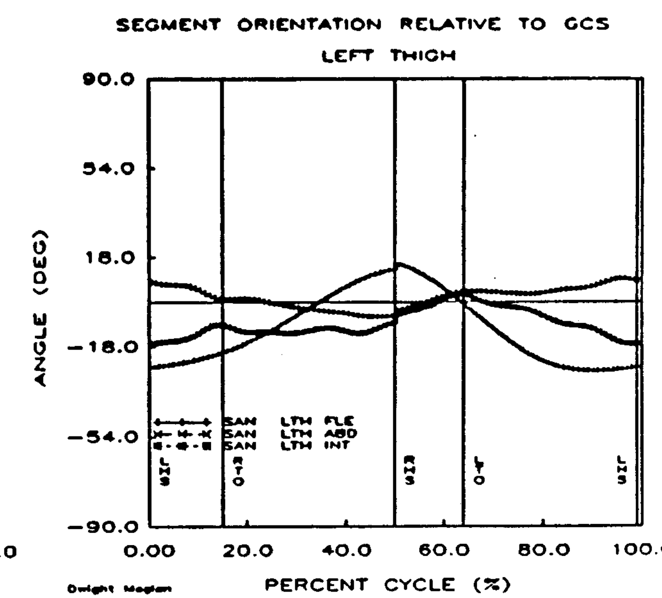

SEGMENT ORIENTATION RELATIVE TO GCS
RIGHT FOOT
ANGLE (DEG)
90.0
54.0
18.0
-18.0
-54.0
-90.0
SAN RFT FLE
SAN RFT ABD
SAN RFT INT
0.00
20.0
40.0
60.0
80.0
100.0
PERCENT CYCLE (%)
Dwight Meglan

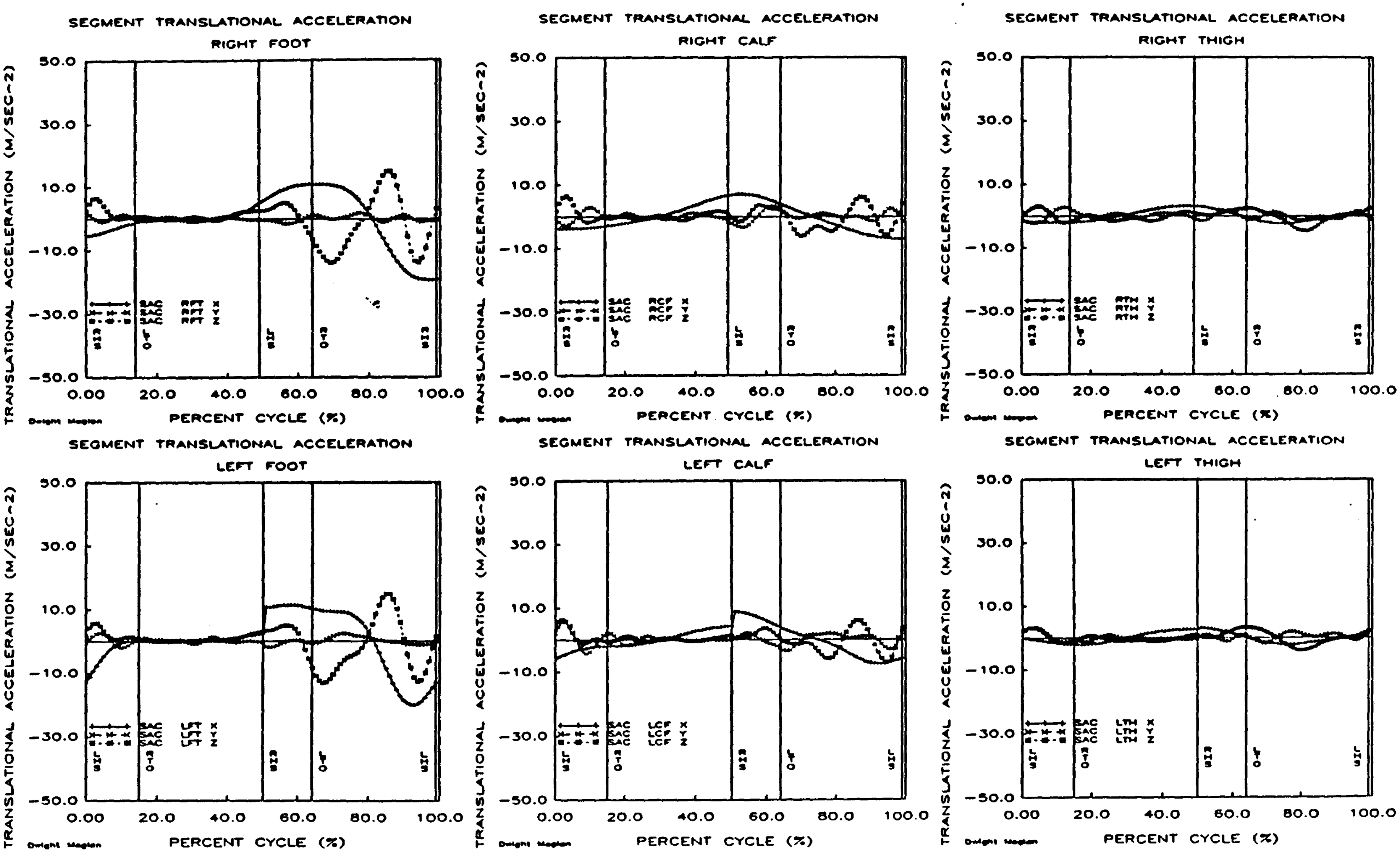

SEGMENT ORIENTATION RELATIVE TO GCS
RIGHT CALF
ANGLE (DEG)
90.0
54.0
18.0
-18.0
-54.0
-90.0
SAN RCF FLE
SAN RCF ABD
SAN RCF INT
0.00
20.0
40.0
60.0
80.0
100.0
PERCENT CYCLE (%)
Dwight Meglan

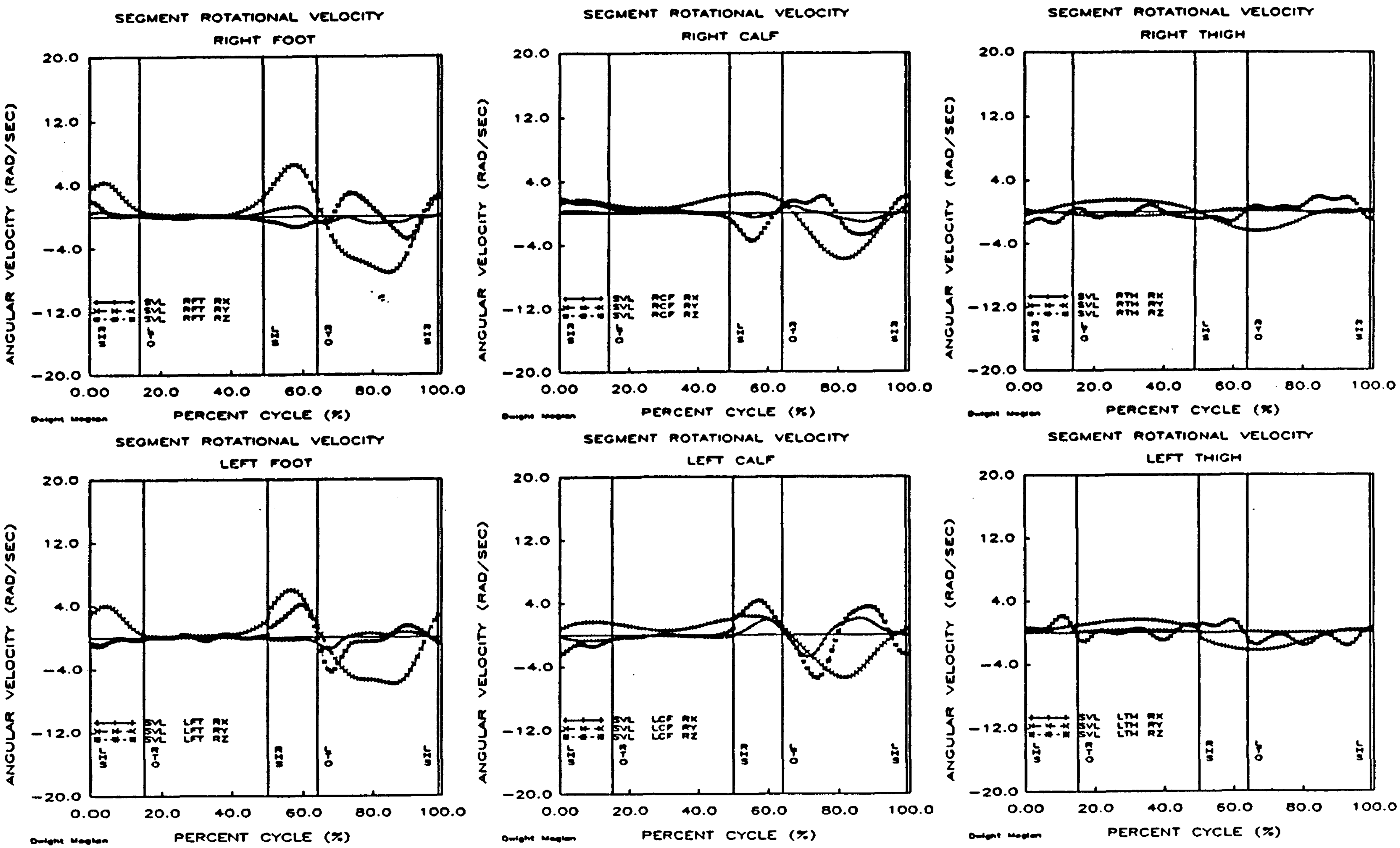

SEGMENT ORIENTATION RELATIVE TO GCS
RIGHT THIGH
ANGLE (DEG)
90.0
54.0
18.0
-18.0
-54.0
-90.0
SAN RTH FLE
SAN RTH ABD
SAN RTH INT
0.00
20.0
40.0
60.0
80.0
100.0
PERCENT CYCLE (%)
Dwight Meglan

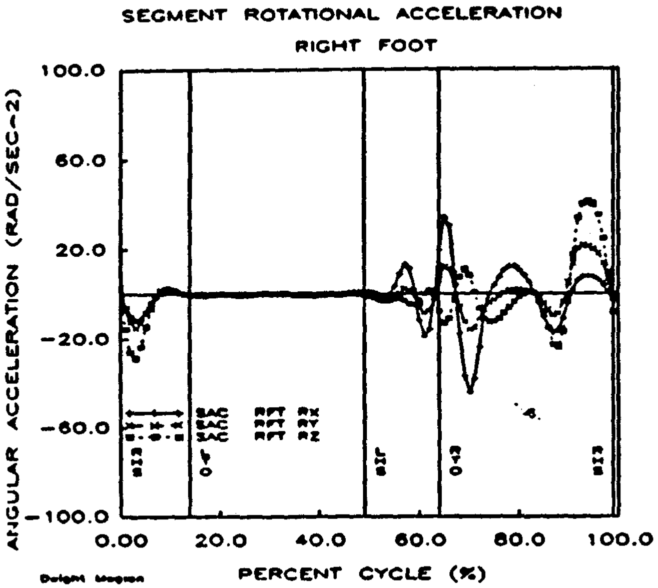

SEGMENT ORIENTATION RELATIVE TO GCS
LEFT FOOT
ANGLE (DEG)
90.0
54.0
18.0
-18.0
-54.0
-90.0
SAN LFT FLE
SAN LFT ABD
SAN LFT INT
0.00
20.0
40.0
60.0
80.0
100.0
PERCENT CYCLE (%)
Dwight Meglan

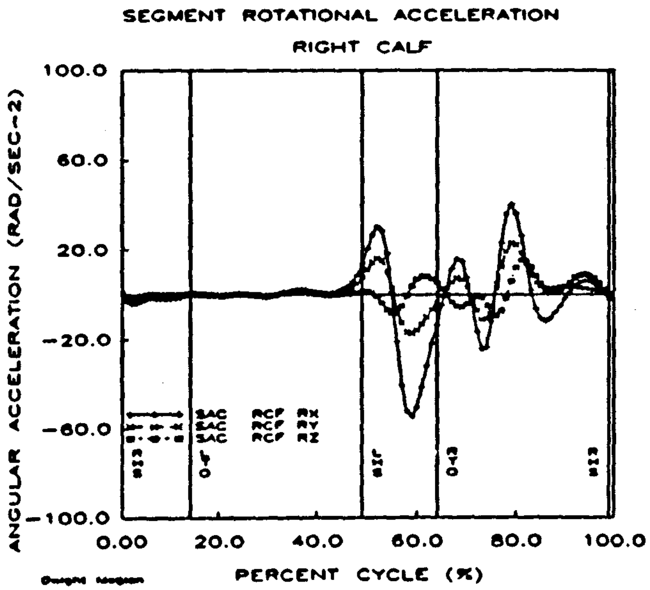

SEGMENT ORIENTATION RELATIVE TO GCS
LEFT CALF
ANGLE (DEG)
90.0
54.0
18.0
-18.0
-54.0
-90.0
SAN LCF FLE
SAN LCF ABD
SAN LCF INT
0.00
20.0
40.0
60.0
80.0
100.0
PERCENT CYCLE (%)
Dwight Meglan

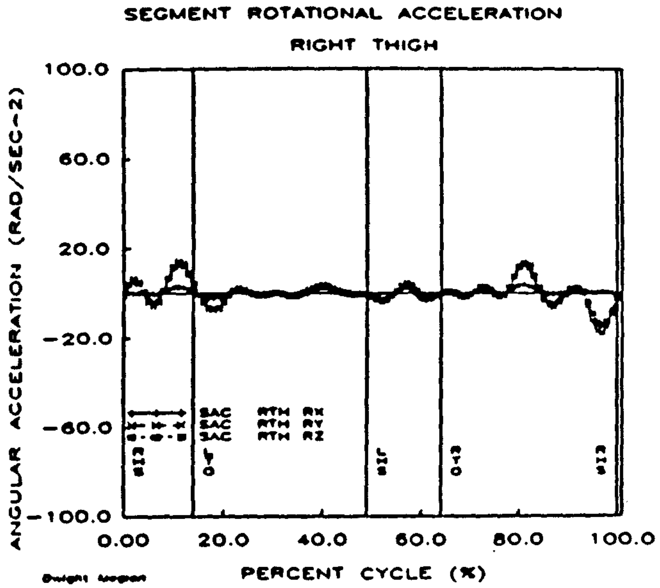

SEGMENT ORIENTATION RELATIVE TO GCS
LEFT THIGH
ANGLE (DEG)
90.0
54.0
18.0
-18.0
-54.0
-90.0
SAN LTH FLE
SAN LTH ABD
SAN LTH INT
0.00
20.0
40.0
60.0
80.0
100.0
PERCENT CYCLE (%)
Dwight Meglan

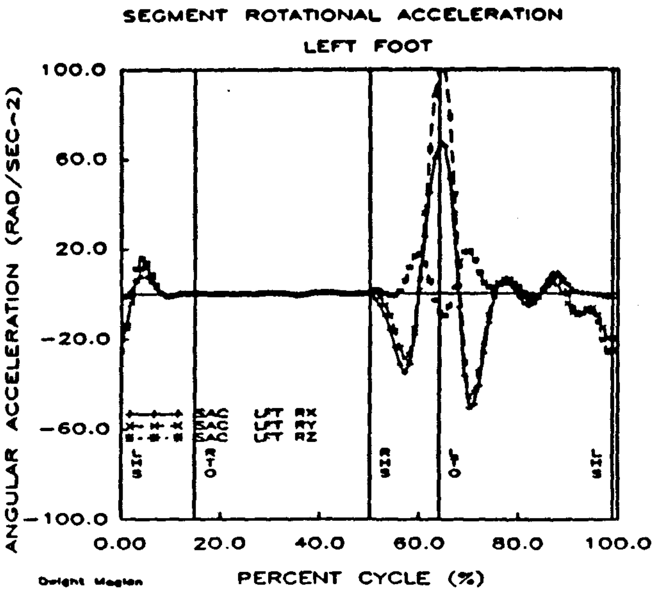

SEGMENT TRANSLATIONAL ACCELERATION
RIGHT FOOT
SEGMENT TRANSLATIONAL ACCELERATION
RIGHT CALF
SEGMENT TRANSLATIONAL ACCELERATION
RIGHT THIGH
SEGMENT TRANSLATIONAL ACCELERATION
LEFT FOOT
SEGMENT TRANSLATIONAL ACCELERATION
LEFT CALF
SEGMENT TRANSLATIONAL ACCELERATION
LEFT THIGH
TRANSLATIONAL ACCELERATION (M/SEC-2)
PERCENT CYCLE (%)
50.0
30.0
10.0
-10.0
-30.0
-50.0
0.00
20.0
40.0
60.0
80.0
100.0
SAC RFT X
SAC RFT Y
SAC RFT Z
SAC RCF X
SAC RCF Y
SAC RCF Z
SAC RTH X
SAC RTH Y
SAC RTH Z
SAC LFT X
SAC LFT Y
SAC LFT Z
SAC LCF X
SAC LCF Y
SAC LCF Z
SAC LTH X
SAC LTH Y
SAC LTH Z
Dwight Meglan

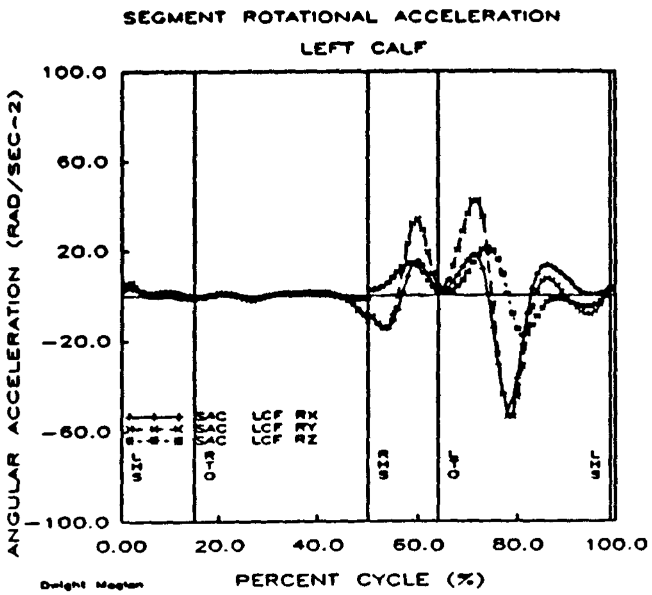

SEGMENT ROTATIONAL VELOCITY
RIGHT FOOT
SEGMENT ROTATIONAL VELOCITY
RIGHT CALF
SEGMENT ROTATIONAL VELOCITY
RIGHT THIGH
SEGMENT ROTATIONAL VELOCITY
LEFT FOOT
SEGMENT ROTATIONAL VELOCITY
LEFT CALF
SEGMENT ROTATIONAL VELOCITY
LEFT THIGH
ANGULAR VELOCITY (RAD/SEC)
PERCENT CYCLE (%)
20.0
12.0
4.0
-4.0
-12.0
-20.0
0.00
20.0
40.0
60.0
80.0
100.0
RCF RX
RCF RY
RCF RZ
RTH RX
RTH RY
RTH RZ
LCF RX
LCF RY
LCF RZ
LTH RX
LTH RY
LTH RZ

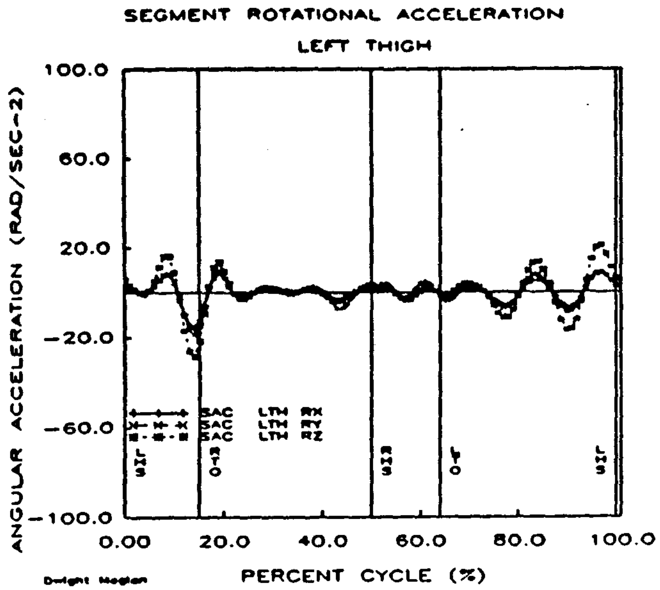

SEGMENT ROTATIONAL ACCELERATION
RIGHT FOOT
ANGULAR ACCELERATION (RAD/SEC-2)
100.0
60.0
20.0
-20.0
-60.0
-100.0
0.00
20.0
40.0
60.0
80.0
100.0
PERCENT CYCLE (%)
Dwight Meglan

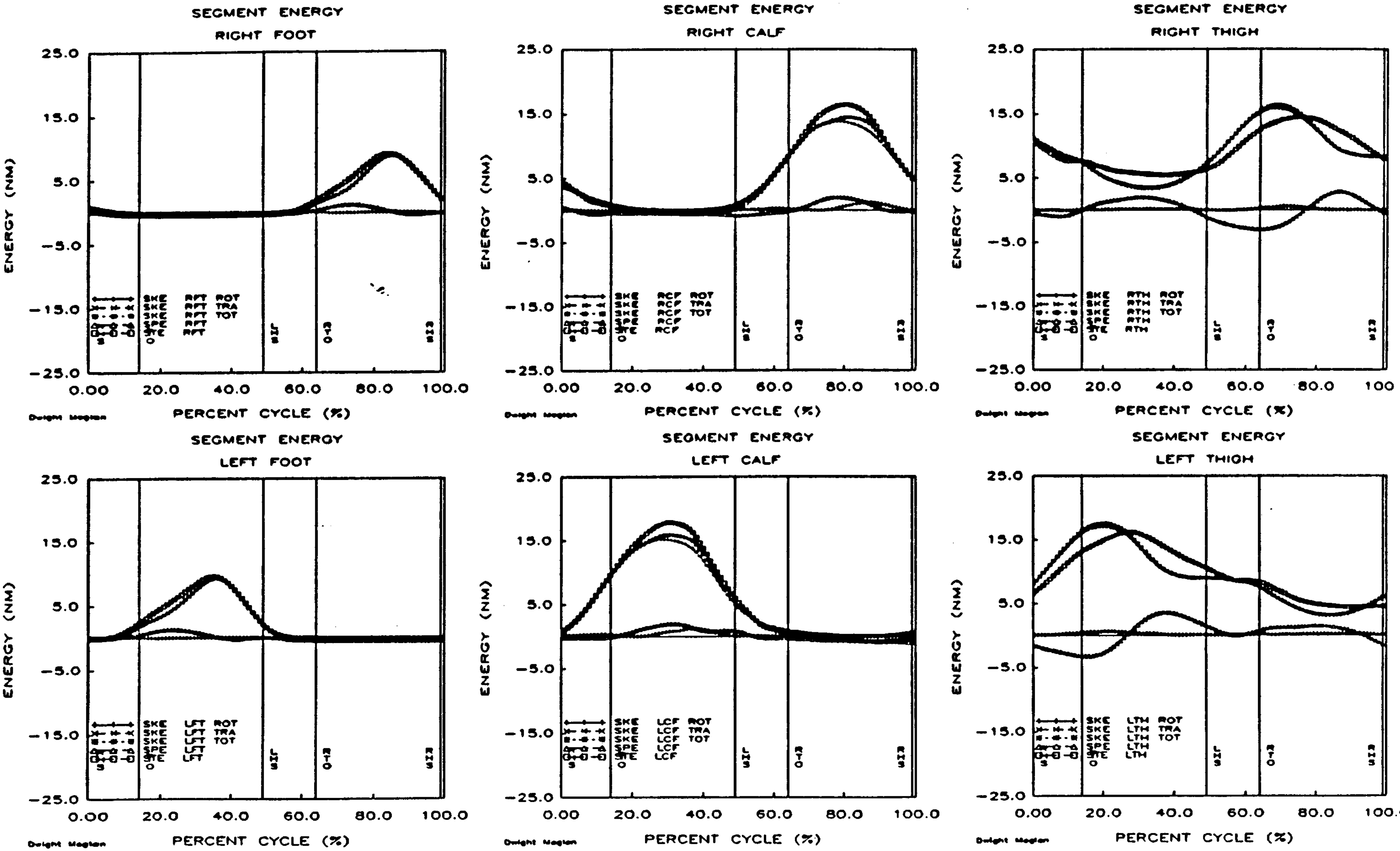

SEGMENT ROTATIONAL ACCELERATION
RIGHT CALF
ANGULAR ACCELERATION (RAD/SEC-2)
100.0
60.0
20.0
-20.0
-60.0
-100.0
0.00
20.0
40.0
60.0
80.0
100.0
PERCENT CYCLE (%)

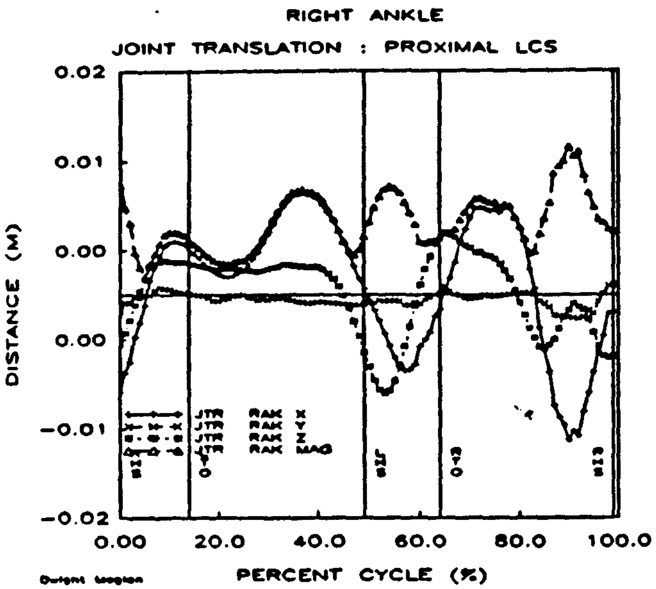

SEGMENT ROTATIONAL ACCELERATION
RIGHT THIGH
ANGULAR ACCELERATION (RAD/SEC-2)
100.0
60.0
20.0
-20.0
-60.0
-100.0
0.00
20.0
40.0
60.0
80.0
100.0
PERCENT CYCLE (%)

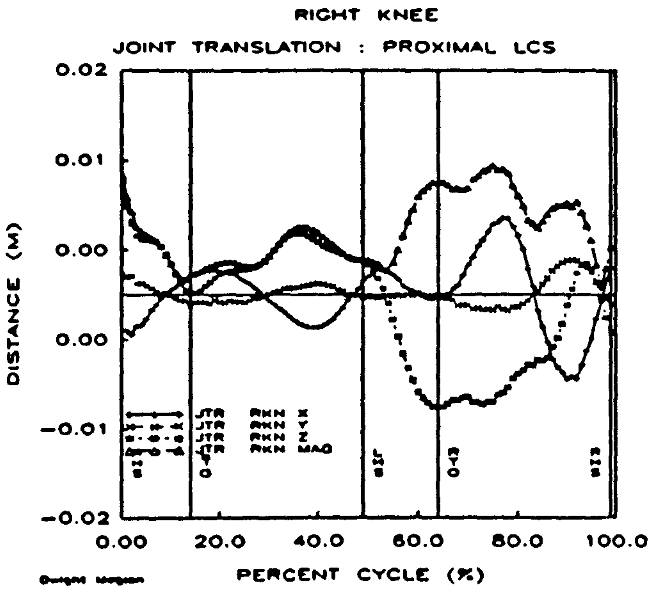

SEGMENT ROTATIONAL ACCELERATION
LEFT FOOT
ANGULAR ACCELERATION (RAD/SEC-2)
100.0
60.0
20.0
-20.0
-60.0
-100.0
0.00
20.0
40.0
60.0
80.0
100.0
PERCENT CYCLE (%)
Dwight Meglan

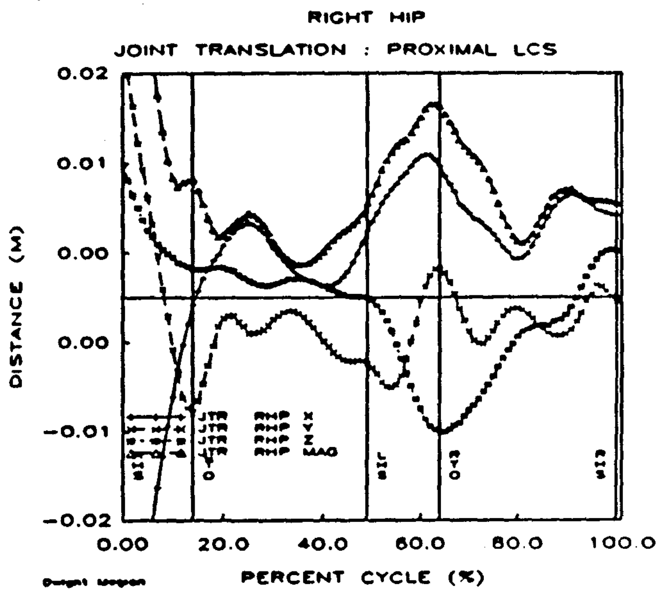

SEGMENT ROTATIONAL ACCELERATION
LEFT CALF
ANGULAR ACCELERATION (RAD/SEC-2)
100.0
60.0
20.0
-20.0
-60.0
-100.0
0.00
20.0
40.0
60.0
80.0
100.0
PERCENT CYCLE (%)

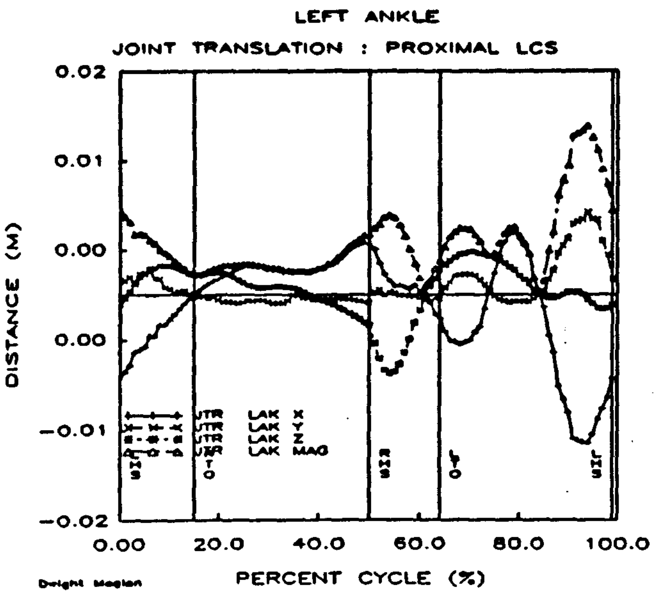

SEGMENT ROTATIONAL ACCELERATION
LEFT THIGH
ANGULAR ACCELERATION (RAD/SEC-2)
100.0
60.0
20.0
-20.0
-60.0
-100.0
0.00
20.0
40.0
60.0
80.0
100.0
PERCENT CYCLE (%)
Dwight Meglan

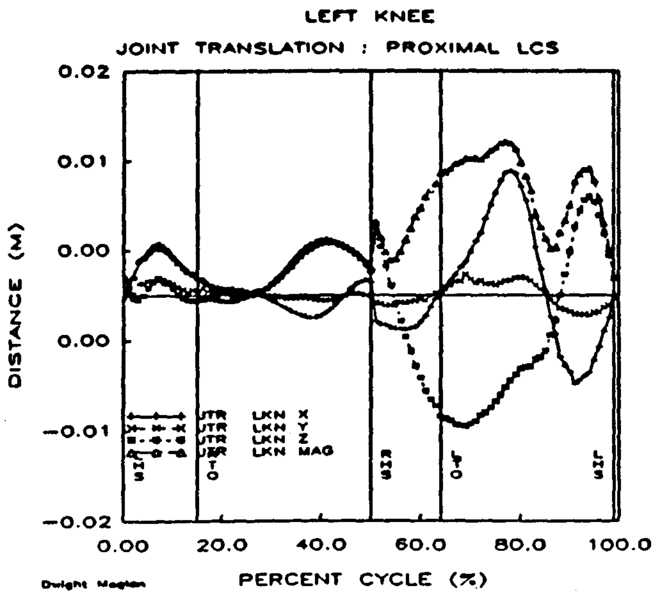

SEGMENT ENERGY
RIGHT FOOT
SEGMENT ENERGY
RIGHT CALF
SEGMENT ENERGY
RIGHT THIGH
SEGMENT ENERGY
LEFT FOOT
SEGMENT ENERGY
LEFT CALF
SEGMENT ENERGY
LEFT THIGH
ENERGY (NM)
PERCENT CYCLE (%)
25.0
15.0
5.0
-5.0
-15.0
-25.0
0.00
20.0
40.0
60.0
80.0
100.0
RFT ROT
RFT TRA
RFT TOT
RCF ROT
RCF TRA
RCF TOT
RTH ROT
RTH TRA
RTH TOT
LFT ROT
LFT TRA
LFT TOT
LCF ROT
LCF TRA
LCF TOT
LTH ROT
LTH TRA
LTH TOT

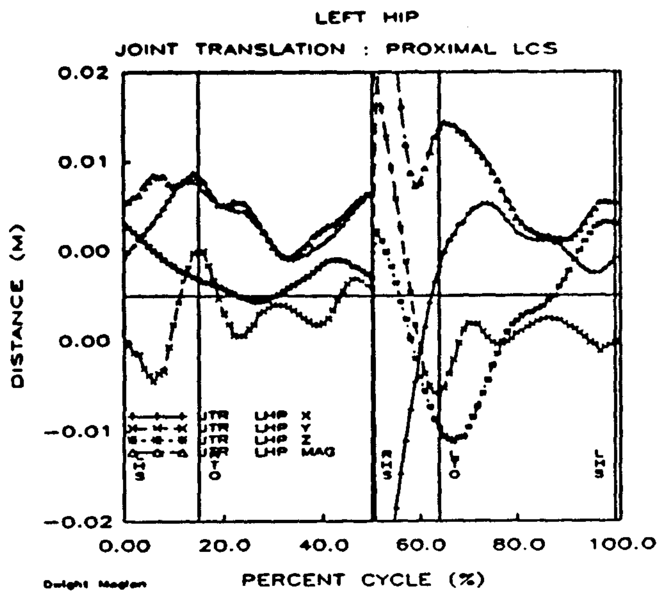

RIGHT ANKLE
JOINT TRANSLATION : PROXIMAL LCS
DISTANCE (M)
PERCENT CYCLE (%)
0.02
0.01
0.00
0.00
-0.01
-0.02
0.00
20.0
40.0
60.0
80.0
100.0

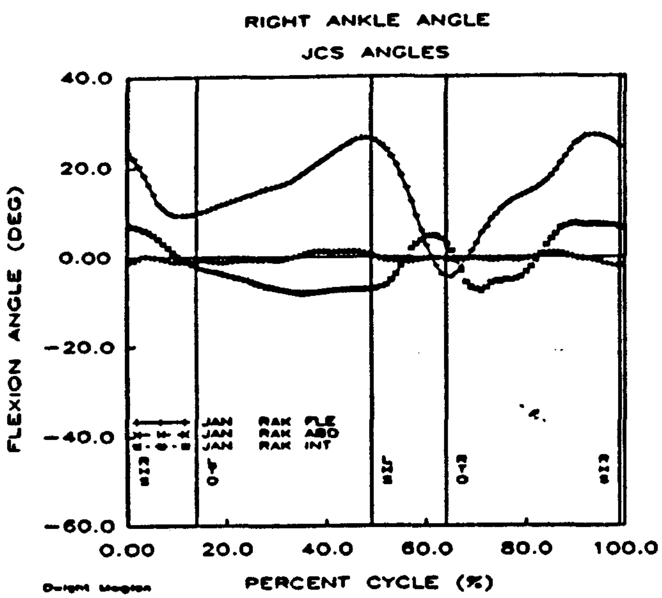

RIGHT KNEE
JOINT TRANSLATION : PROXIMAL LCS
DISTANCE (M)
PERCENT CYCLE (%)
0.02
0.01
0.00
0.00
-0.01
-0.02
0.00
20.0
40.0
60.0
80.0
100.0

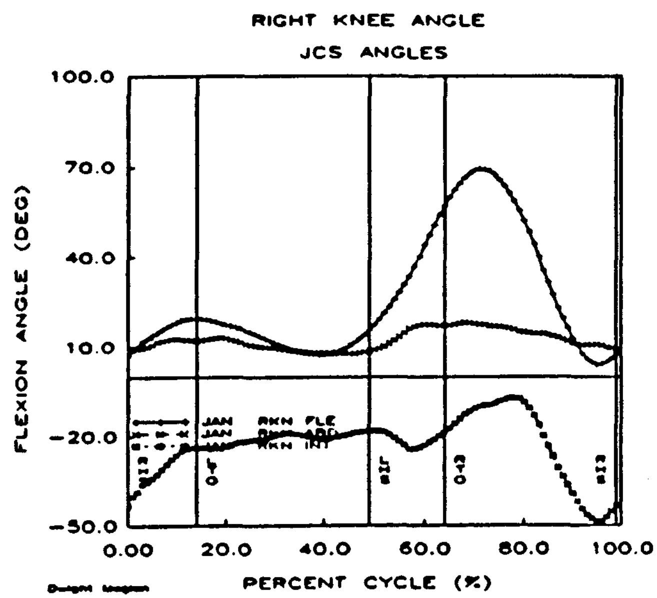

RIGHT HIP
JOINT TRANSLATION : PROXIMAL LCS
DISTANCE (M)
PERCENT CYCLE (%)
0.02
0.01
0.00
0.00
-0.01
-0.02
0.00
20.0
40.0
60.0
80.0
100.0

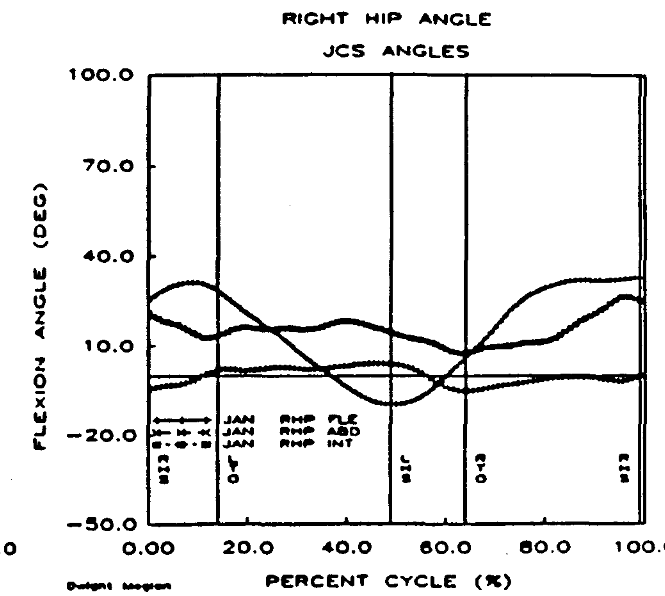

LEFT ANKLE
JOINT TRANSLATION : PROXIMAL LCS
DISTANCE (M)
PERCENT CYCLE (%)
0.02
0.01
0.00
0.00
-0.01
-0.02
0.00
20.0
40.0
60.0
80.0
100.0

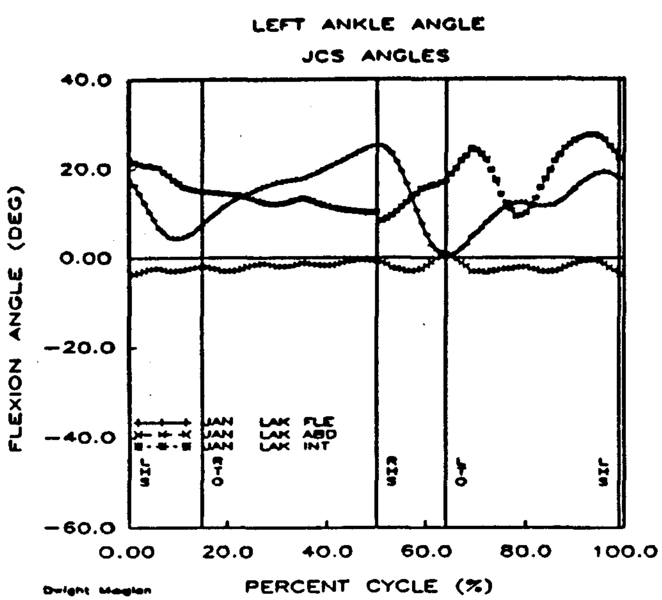

LEFT KNEE
JOINT TRANSLATION : PROXIMAL LCS
DISTANCE (M)
PERCENT CYCLE (%)
0.02
0.01
0.00
0.00
-0.01
-0.02
0.00
20.0
40.0
60.0
80.0
100.0

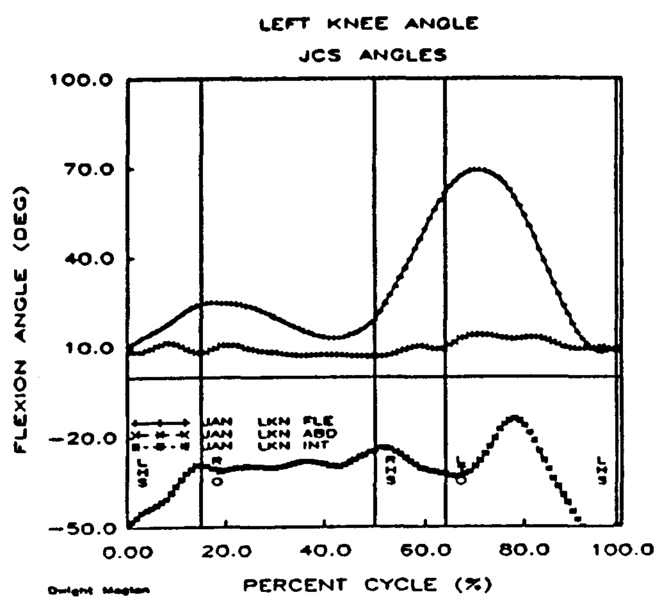

LEFT HIP
JOINT TRANSLATION : PROXIMAL LCS
DISTANCE (M)
PERCENT CYCLE (%)
0.02
0.01
0.00
0.00
-0.01
-0.02
0.00
20.0
40.0
60.0
80.0
100.0

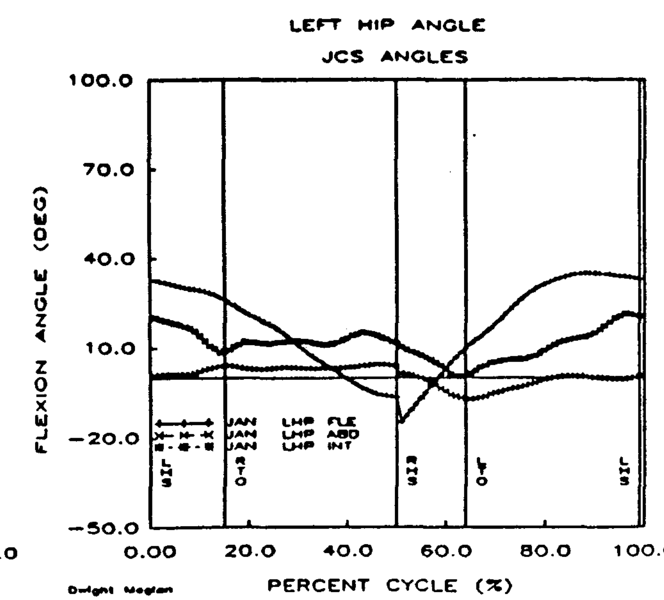

RIGHT ANKLE ANGLE
JCS ANGLES
FLEXION ANGLE (DEG)
PERCENT CYCLE (%)
JAN RAK FLE
JAN RAK ABD
JAN RAK INT
Dwight Meglan

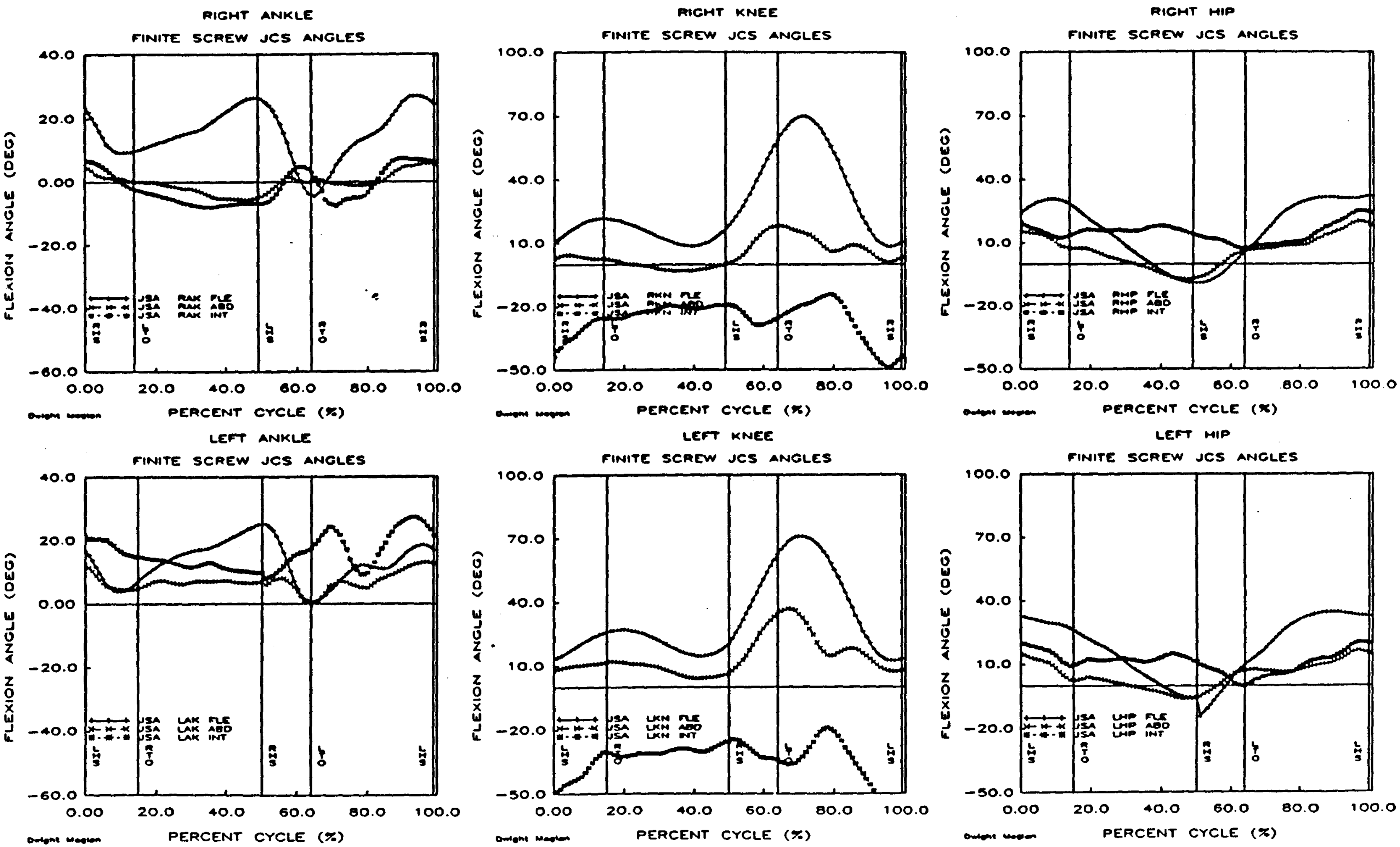

RIGHT KNEE ANGLE
JCS ANGLES
FLEXION ANGLE (DEG)
PERCENT CYCLE (%)
JAN RKN FLE
JAN RKN ABD
JAN RKN INT
Dwight Meglan

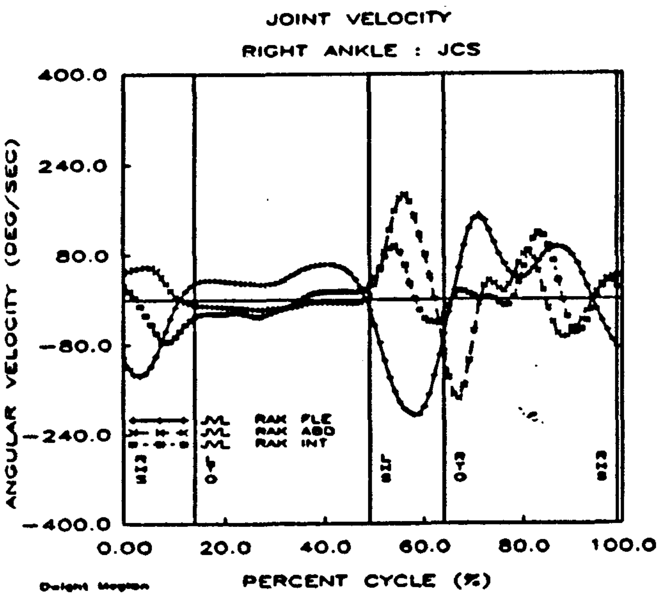

RIGHT HIP ANGLE
JCS ANGLES
FLEXION ANGLE (DEG)
PERCENT CYCLE (%)
JAN RHP FLE
JAN RHP ABD
JAN RHP INT
Dwight Meglan

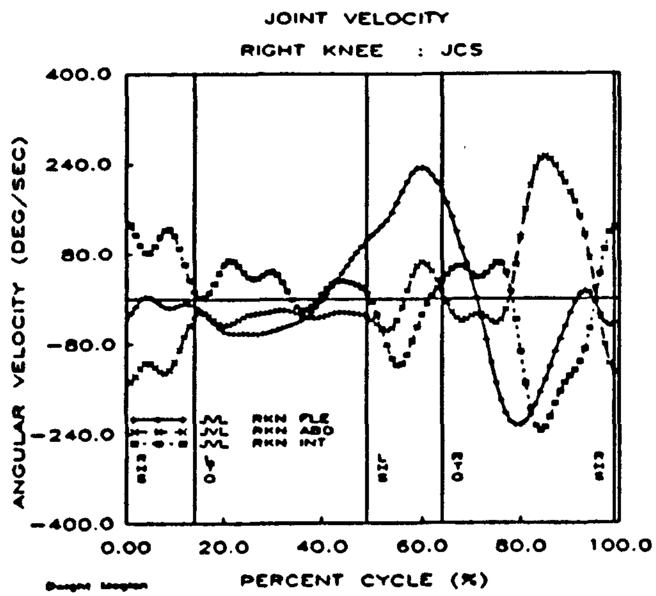

LEFT ANKLE ANGLE
JCS ANGLES
FLEXION ANGLE (DEG)
PERCENT CYCLE (%)
JAN LAK FLE
JAN LAK ABD
JAN LAK INT
Dwight Meglan

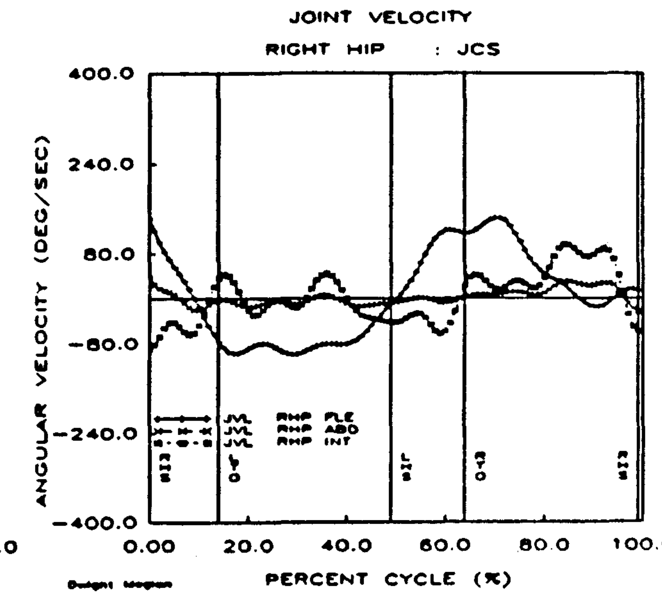

LEFT KNEE ANGLE
JCS ANGLES
FLEXION ANGLE (DEG)
PERCENT CYCLE (%)
JAN LKN FLE
JAN LKN ABD
JAN LKN INT
Dwight Meglan

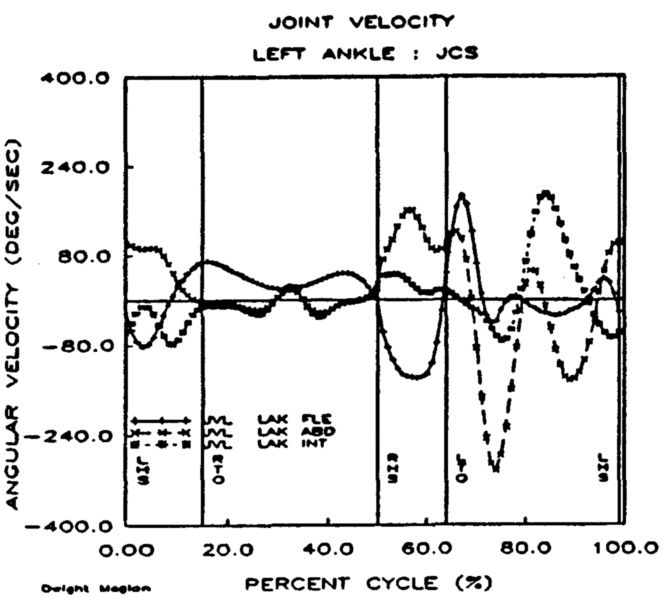

LEFT HIP ANGLE
JCS ANGLES
FLEXION ANGLE (DEG)
PERCENT CYCLE (%)
JAN LHP FLE
JAN LHP ABD
JAN LHP INT
Dwight Meglan

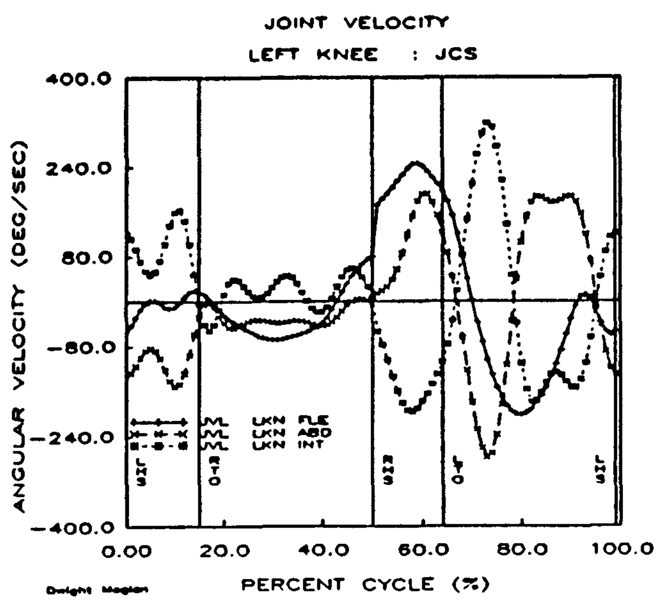

RIGHT ANKLE
FINITE SCREW JCS ANGLES
JSA RAK FLE
JSA RAK ABD
JSA RAK INT
RIGHT KNEE
FINITE SCREW JCS ANGLES
RIGHT HIP
FINITE SCREW JCS ANGLES
JSA RHP FLE
JSA RHP ABD
JSA RHP INT
LEFT ANKLE
FINITE SCREW JCS ANGLES
JSA LAK FLE
JSA LAK ABD
JSA LAK INT
LEFT KNEE
FINITE SCREW JCS ANGLES
JSA LKN FLE
JSA LKN ABD
JSA LKN INT
LEFT HIP
FINITE SCREW JCS ANGLES
JSA LHP FLE
JSA LHP ABD
JSA LHP INT
FLEXION ANGLE (DEG)
PERCENT CYCLE (%)
Dwight Meglan

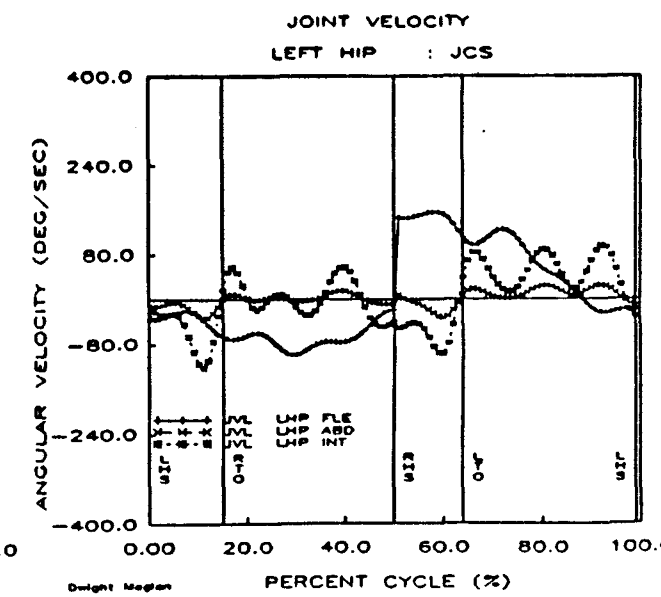

JOINT VELOCITY
RIGHT ANKLE : JCS
ANGULAR VELOCITY (DEG/SEC)
400.0
240.0
80.0
−80.0
−240.0
−400.0
0.00
20.0
40.0
60.0
80.0
100.0
PERCENT CYCLE (%)
JVL RAK FLE
JVL RAK ABD
JVL RAK INT
Dwight Meglan

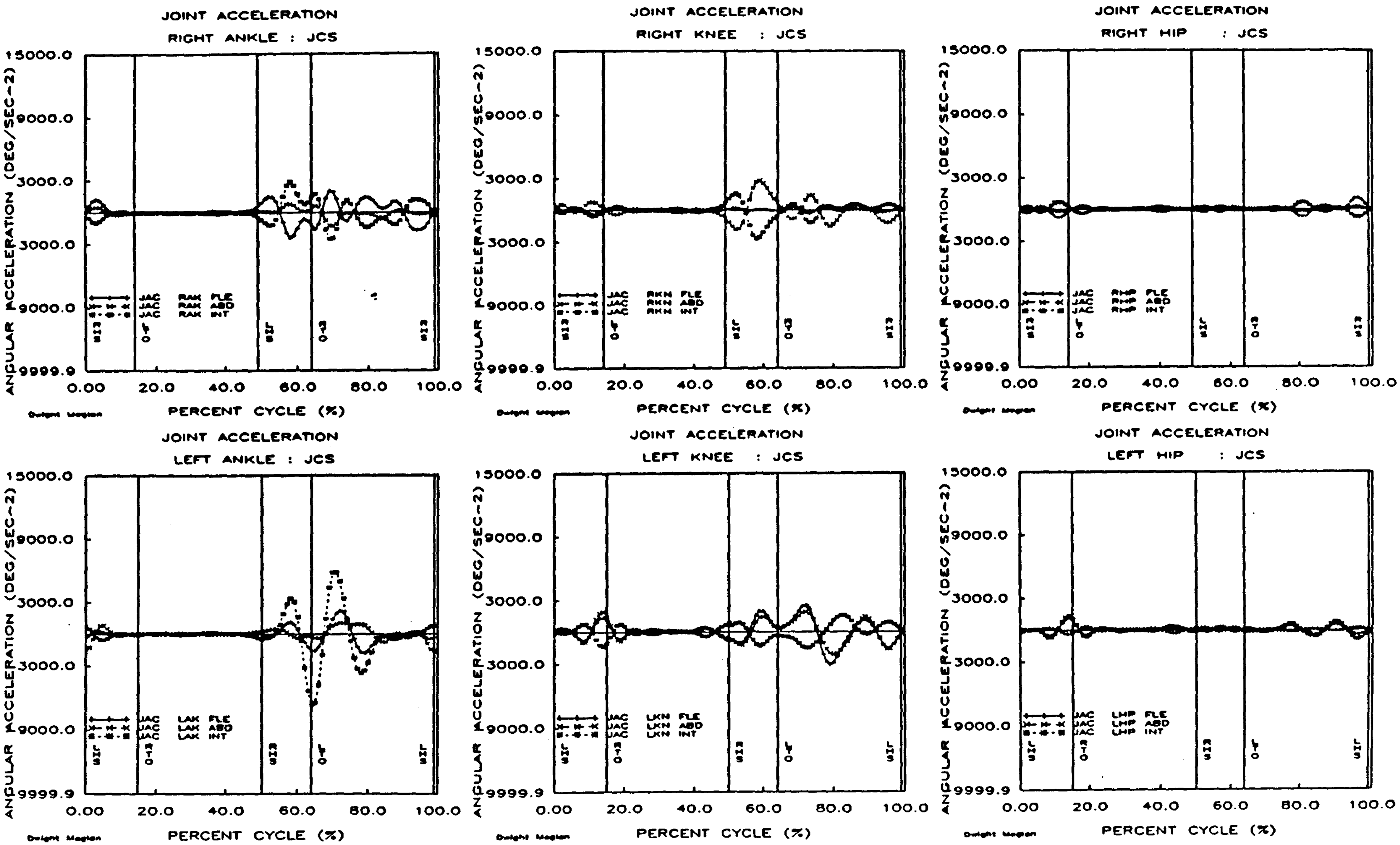

JOINT VELOCITY
RIGHT KNEE : JCS
ANGULAR VELOCITY (DEG/SEC)
400.0
240.0
80.0
−80.0
−240.0
−400.0
0.00
20.0
40.0
60.0
80.0
100.0
PERCENT CYCLE (%)
JVL RKN FLE
JVL RKN ABD
JVL RKN INT
Dwight Meglan

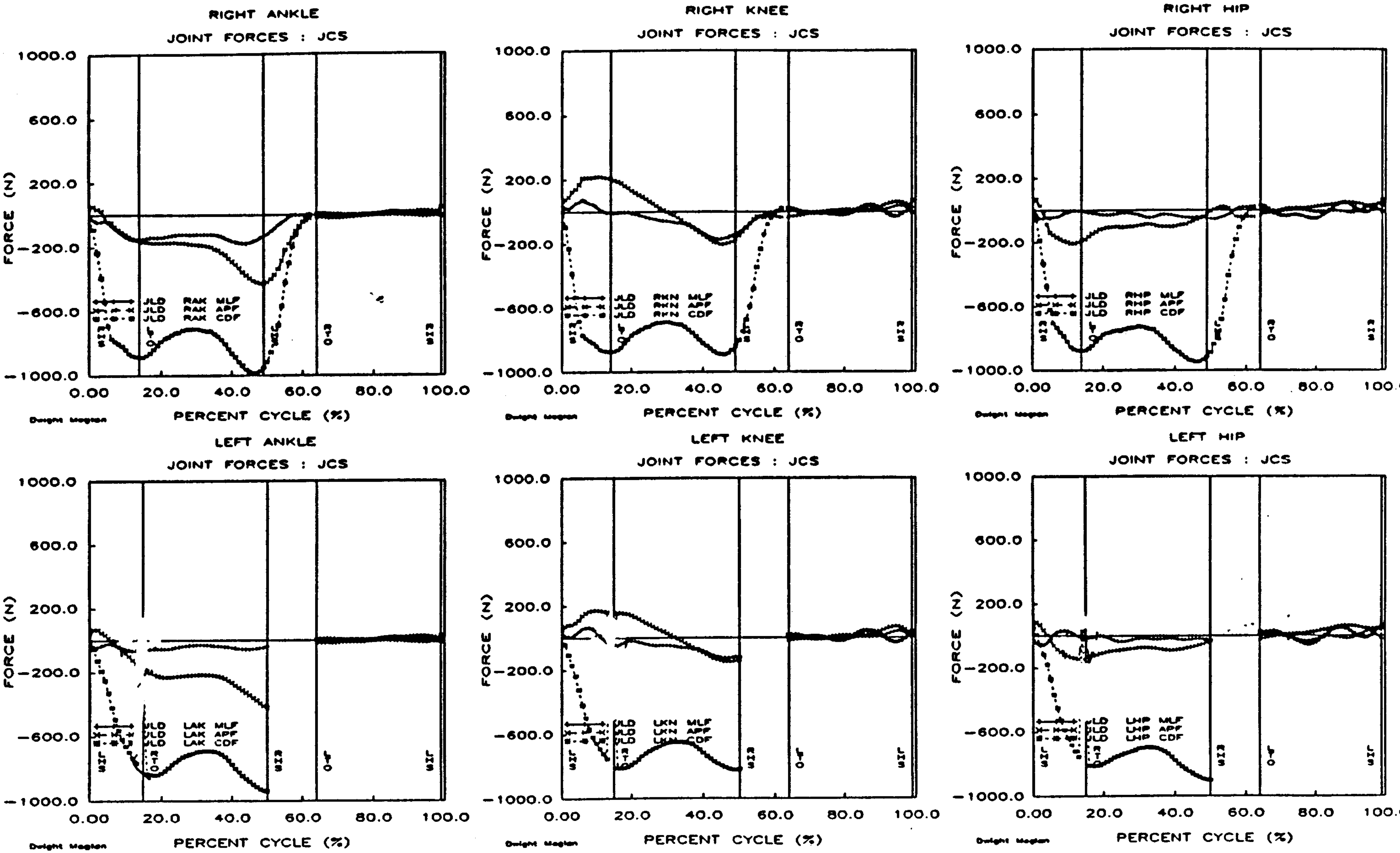

JOINT VELOCITY
RIGHT HIP : JCS
ANGULAR VELOCITY (DEG/SEC)
400.0
240.0
80.0
−80.0
−240.0
−400.0
0.00
20.0
40.0
60.0
80.0
100.0
PERCENT CYCLE (%)
JVL RHP FLE
JVL RHP ABD
JVL RHP INT
Dwight Meglan

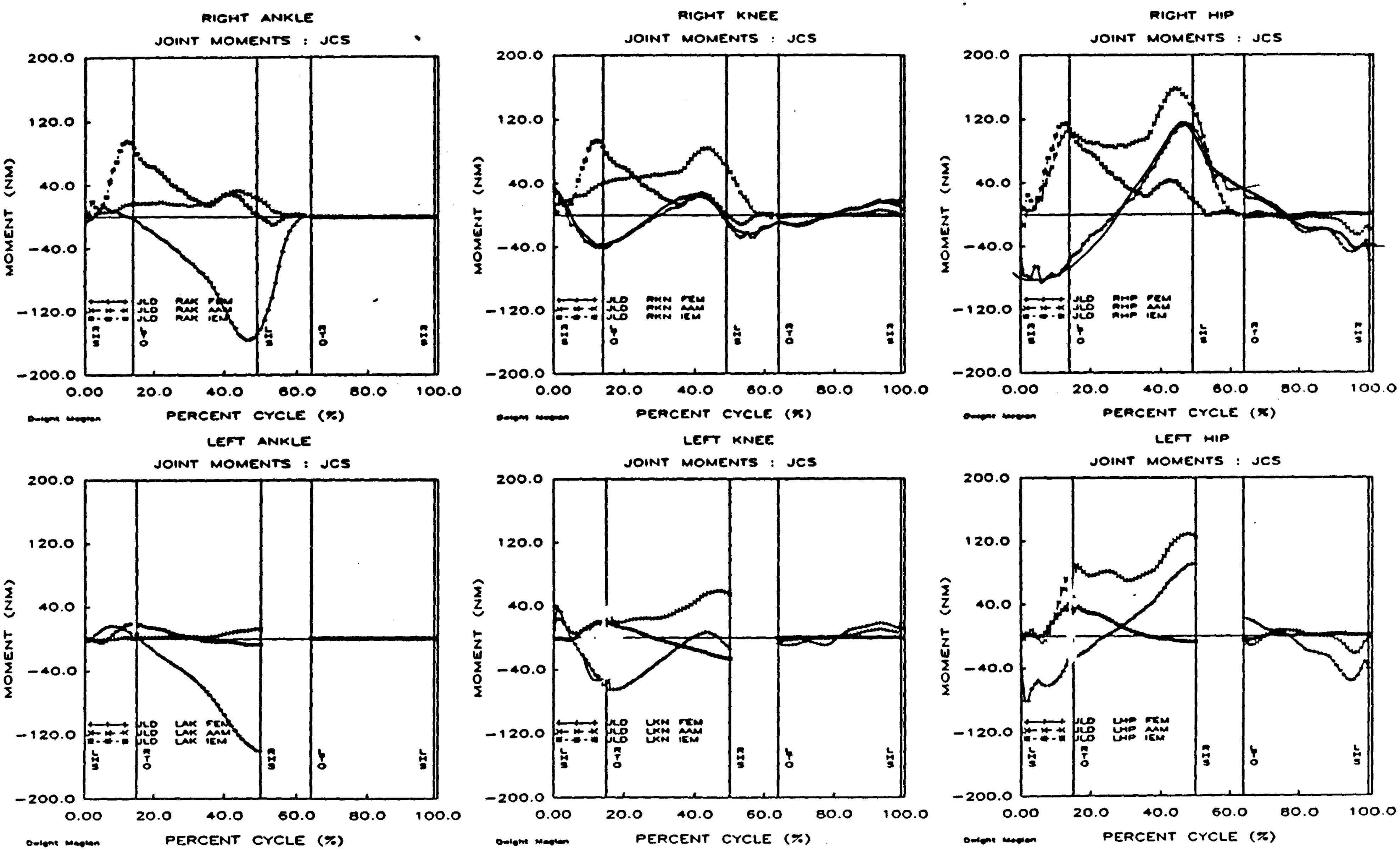

JOINT VELOCITY
LEFT ANKLE : JCS
ANGULAR VELOCITY (DEG/SEC)
400.0
240.0
80.0
−80.0
−240.0
−400.0
0.00
20.0
40.0
60.0
80.0
100.0
PERCENT CYCLE (%)
JVL LAK FLE
JVL LAK ABD
JVL LAK INT
Dwight Meglan


JOINT VELOCITY
LEFT KNEE : JCS
ANGULAR VELOCITY (DEG/SEC)
400.0
240.0
80.0
−80.0
−240.0
−400.0
0.00
20.0
40.0
60.0
80.0
100.0
PERCENT CYCLE (%)
JVL LKN FLE
JVL LKN ABD
JVL LKN INT
Dwight Meglan


JOINT VELOCITY
LEFT HIP : JCS
ANGULAR VELOCITY (DEG/SEC)
400.0
240.0
80.0
−80.0
−240.0
−400.0
0.00
20.0
40.0
60.0
80.0
100.0
PERCENT CYCLE (%)
JVL LHP FLE
JVL LHP ABD
JVL LHP INT
Dwight Meglan

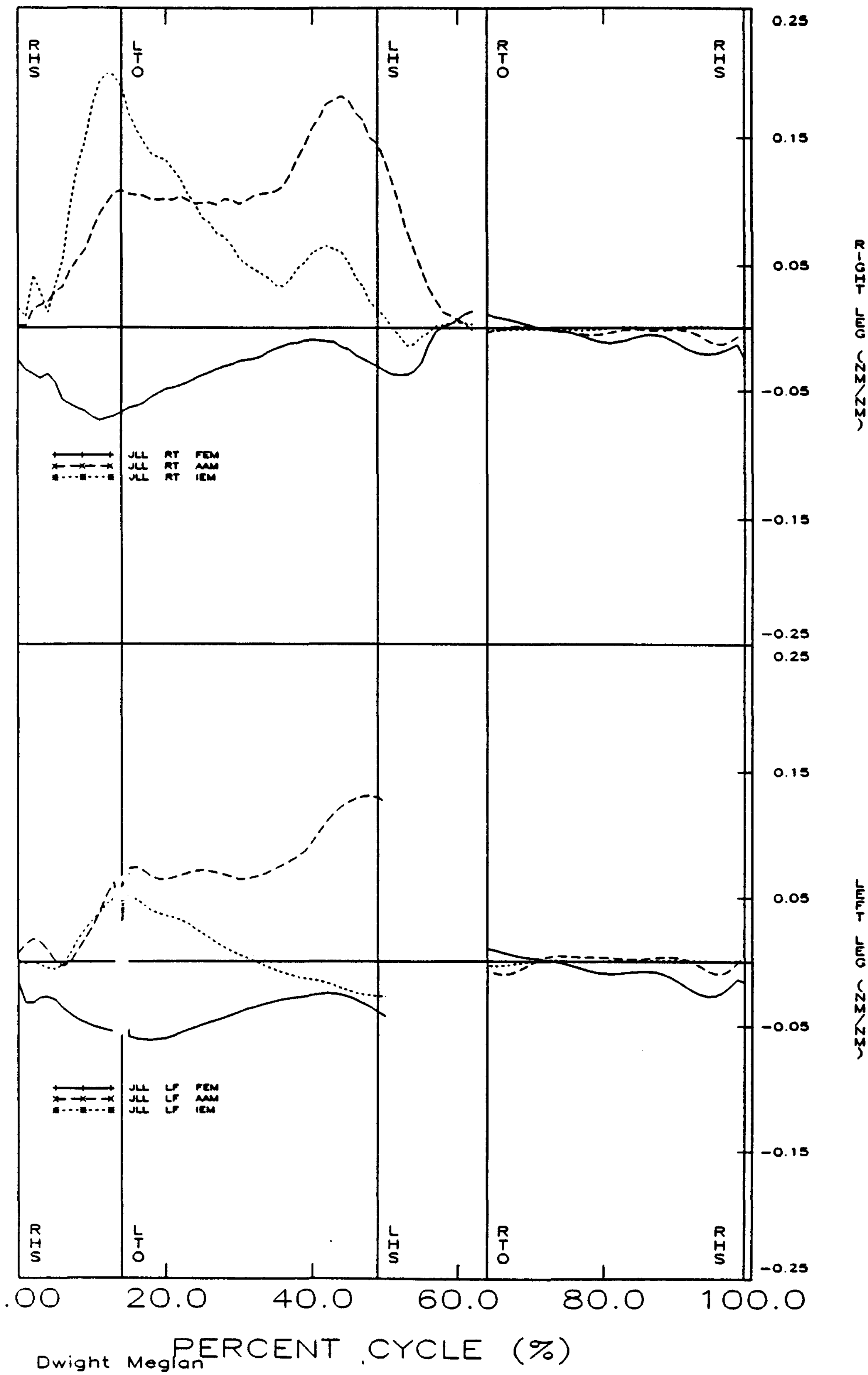

JOINT ACCELERATION
RIGHT ANKLE : JCS
JAC RAK FLE
JAC RAK ABD
JAC RAK INT
JOINT ACCELERATION
RIGHT KNEE : JCS
JAC RKN FLE
JAC RKN ABD
JAC RKN INT
JOINT ACCELERATION
RIGHT HIP : JCS
JAC RHP FLE
JAC RHP ABD
JAC RHP INT
JOINT ACCELERATION
LEFT ANKLE : JCS
JAC LAK FLE
JAC LAK ABD
JAC LAK INT
JOINT ACCELERATION
LEFT KNEE : JCS
JAC LKN FLE
JAC LKN ABD
JAC LKN INT
JOINT ACCELERATION
LEFT HIP : JCS
JAC LHP FLE
JAC LHP ABD
JAC LHP INT
ANGULAR ACCELERATION (DEG/SEC-2)
15000.0
9000.0
3000.0
3000.0
9000.0
9999.9
0.00
20.0
40.0
60.0
80.0
100.0
PERCENT CYCLE (%)
Dwight Meglan


RIGHT ANKLE
JOINT FORCES : JCS
JLD RAK MLF
JLD RAK APF
JLD RAK CDF
RIGHT KNEE
JOINT FORCES : JCS
JLD RKN MLF
JLD RKN APF
JLD RKN CDF
RIGHT HIP
JOINT FORCES : JCS
JLD RHP MLF
JLD RHP APF
JLD RHP CDF
LEFT ANKLE
JOINT FORCES : JCS
JLD LAK MLF
JLD LAK APF
JLD LAK CDF
LEFT KNEE
JOINT FORCES : JCS
JLD LKN MLF
JLD LKN APF
JLD LKN CDF
LEFT HIP
JOINT FORCES : JCS
JLD LHP MLF
JLD LHP APF
JLD LHP CDF
FORCE (N)
1000.0
600.0
200.0
−200.0
−600.0
−1000.0
PERCENT CYCLE (%)
0.00
20.0
40.0
60.0
80.0
100.0



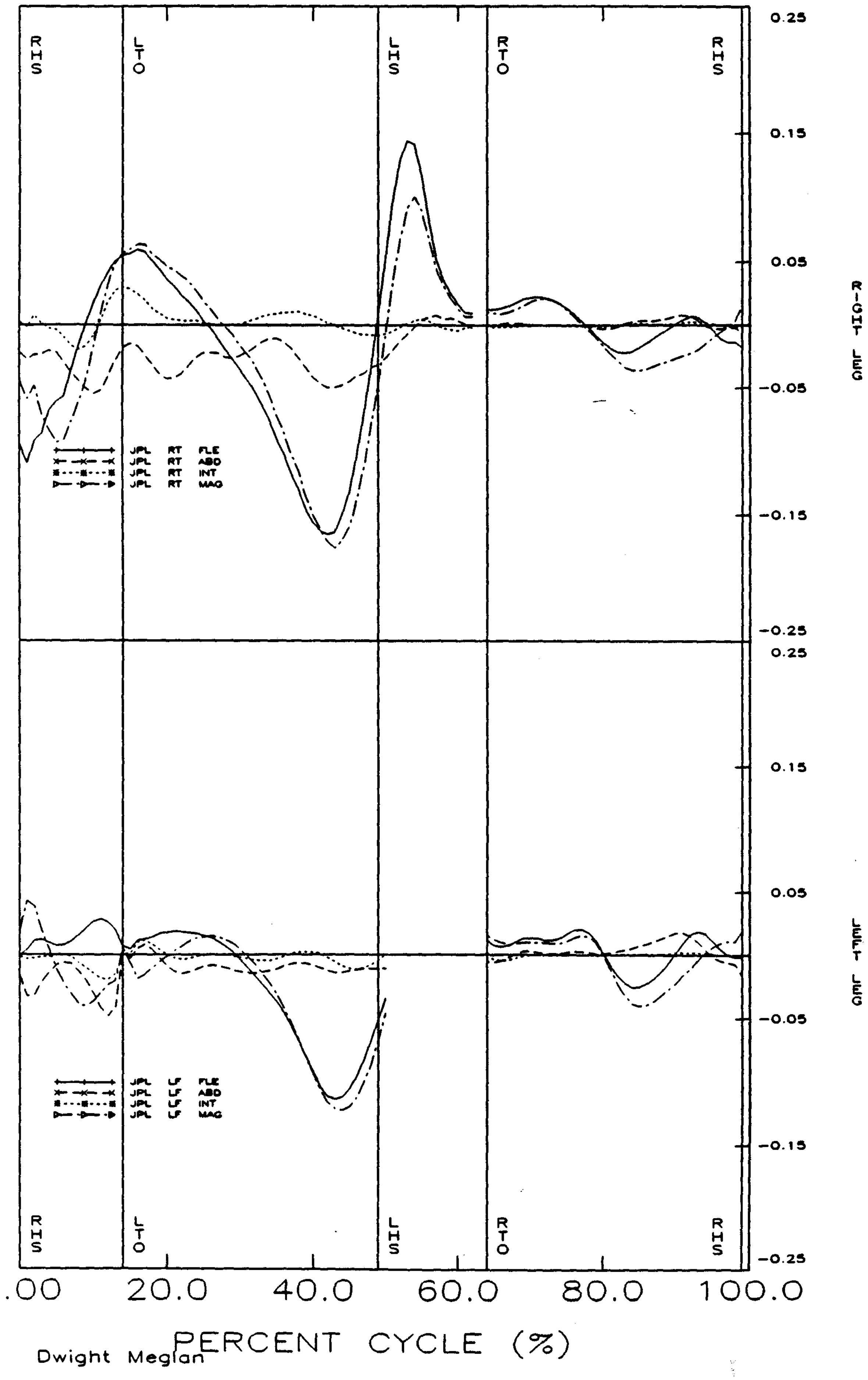
RIGHT ANKLE
JOINT MOMENTS : JCS
MOMENT (NM)
PERCENT CYCLE (%)
JLD RAK FEM
JLD RAK AAM
JLD RAK IEM
RIGHT KNEE
JOINT MOMENTS : JCS
MOMENT (NM)
PERCENT CYCLE (%)
JLD RKN FEM
JLD RKN AAM
JLD RKN IEM
RIGHT HIP
JOINT MOMENTS : JCS
MOMENT (NM)
PERCENT CYCLE (%)
JLD RHP FEM
JLD RHP AAM
JLD RHP IEM
LEFT ANKLE
JOINT MOMENTS : JCS
MOMENT (NM)
PERCENT CYCLE (%)
JLD LAK FEM
JLD LAK AAM
JLD LAK IEM
LEFT KNEE
JOINT MOMENTS : JCS
MOMENT (NM)
PERCENT CYCLE (%)
JLD LKN FEM
JLD LKN AAM
JLD LKN IEM
LEFT HIP
JOINT MOMENTS : JCS
MOMENT (NM)
PERCENT CYCLE (%)
JLD LHP FEM
JLD LHP AAM
JLD LHP IEM

MMED LEG MOMENTS NORMWTHT : JCS



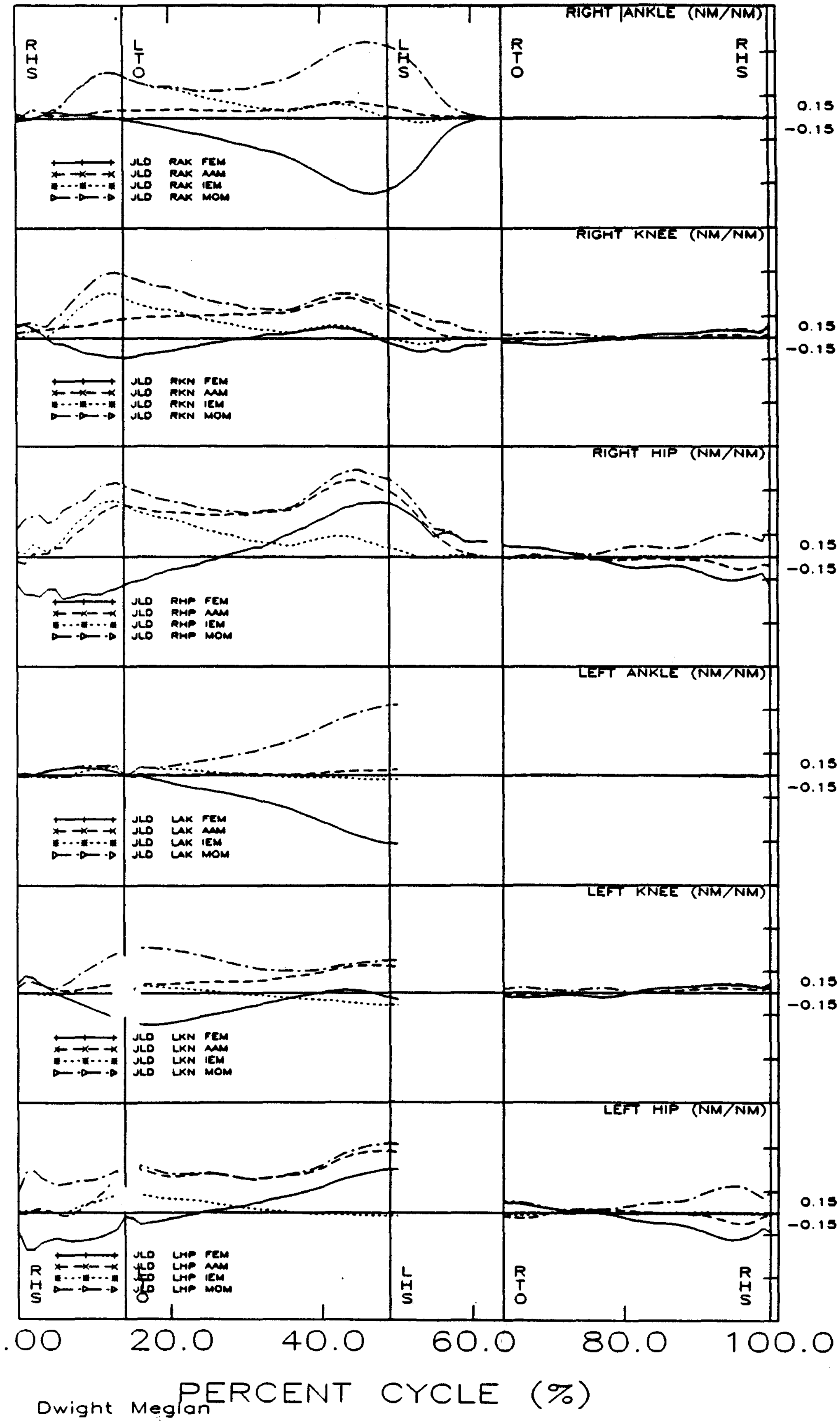
Dwight Meglan

.UMMED LEG POWER NORMWTHT : JCS



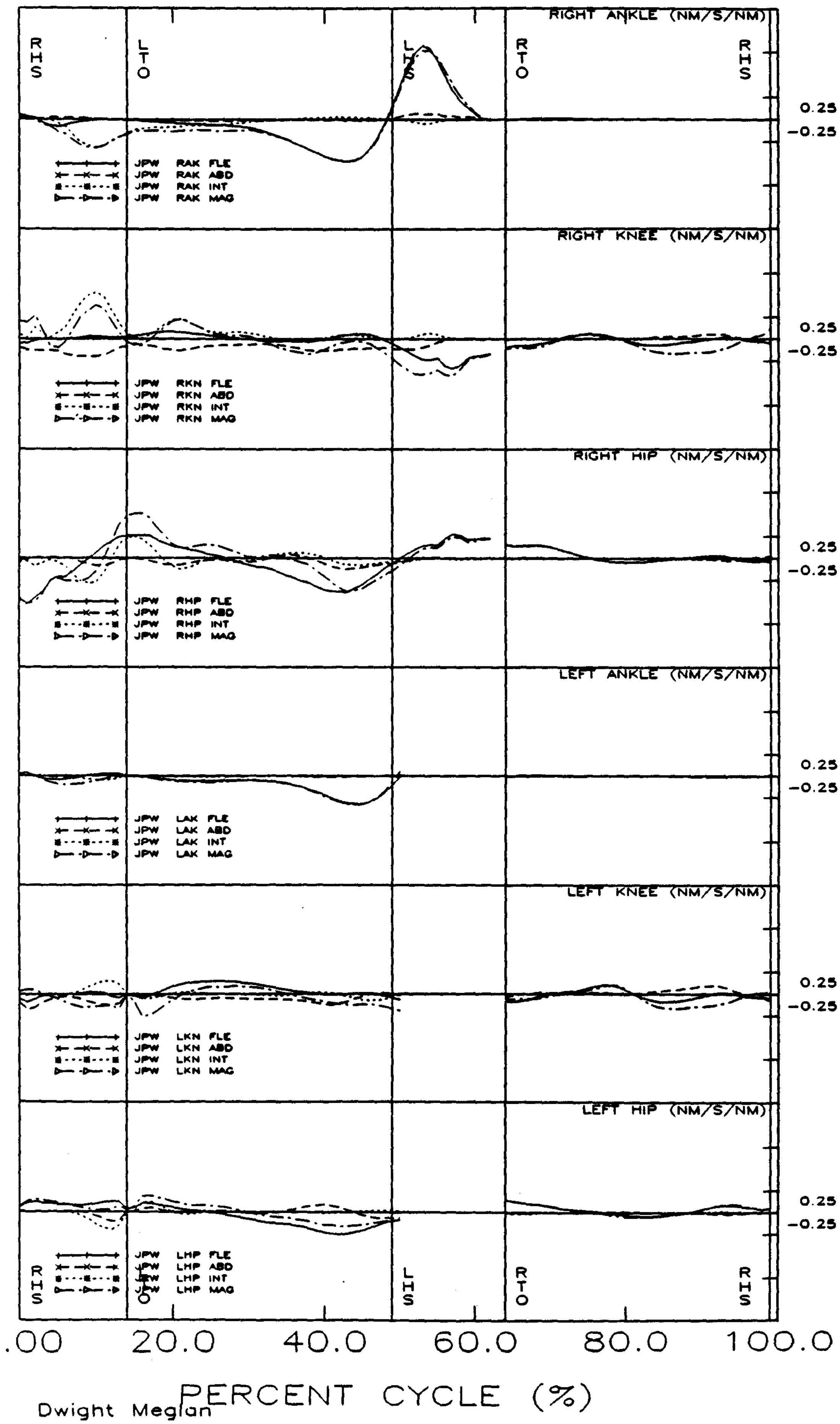
Dwight Meglan

JOINT MOMENTS NORMWTHT : JCS

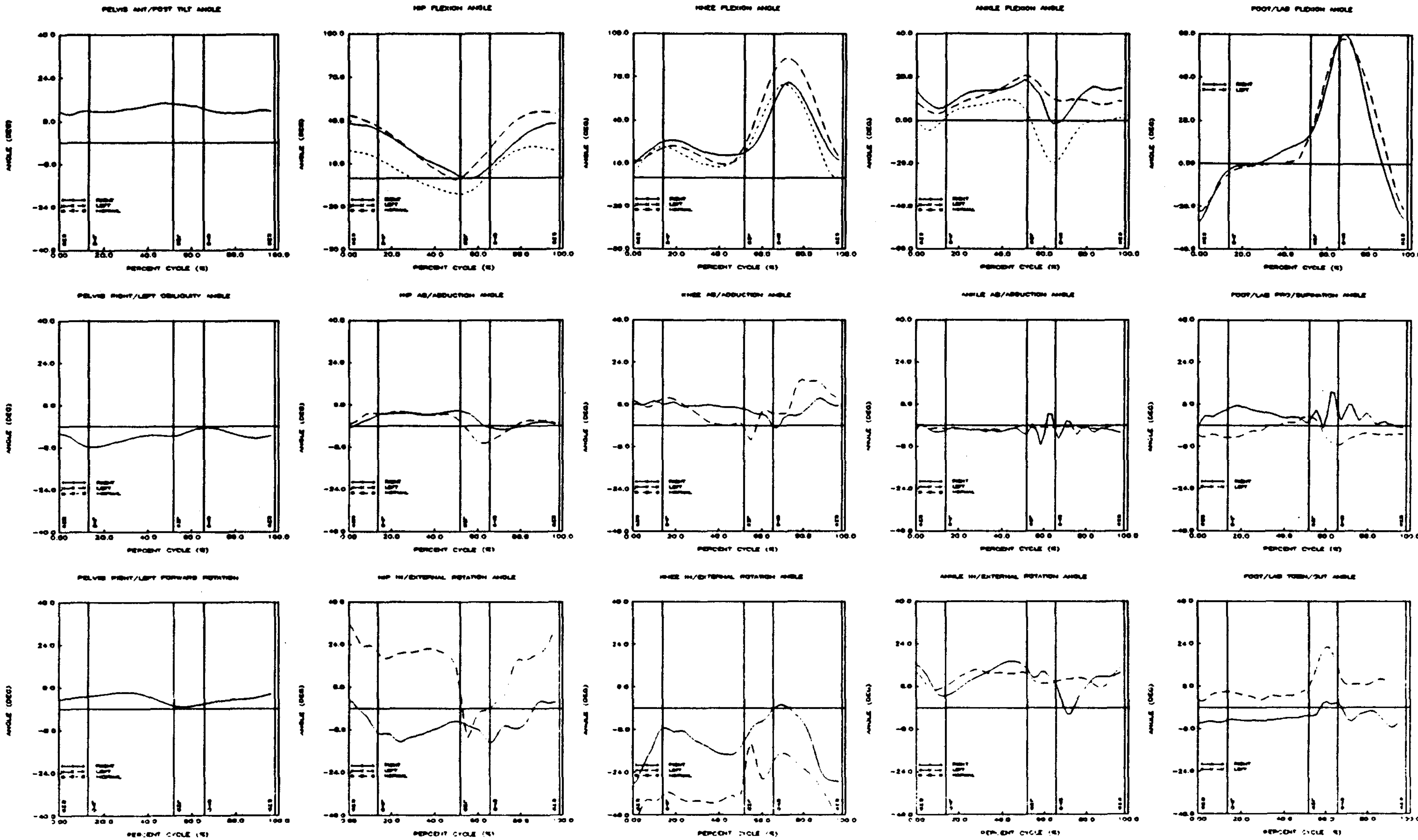


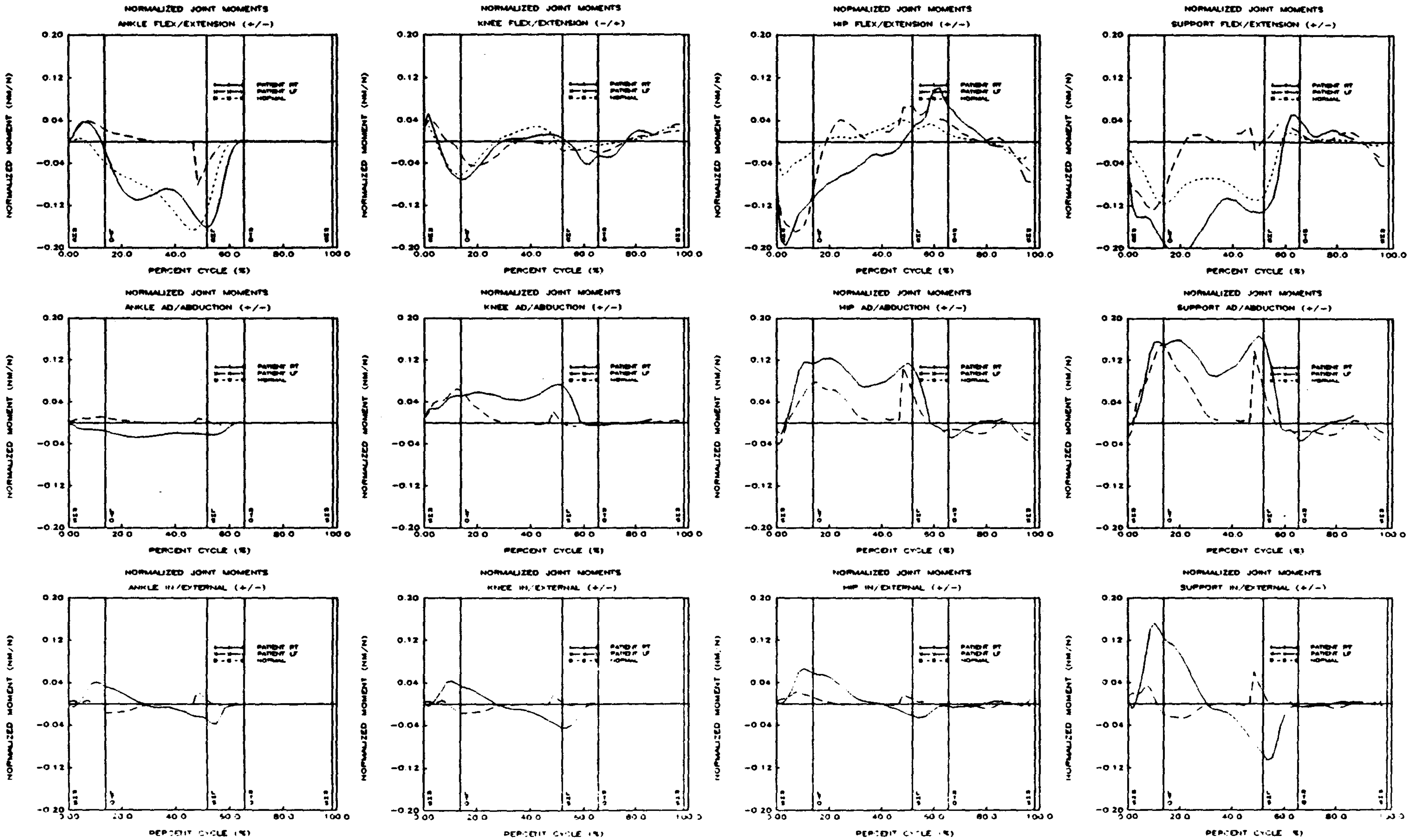
Dwight Meglan

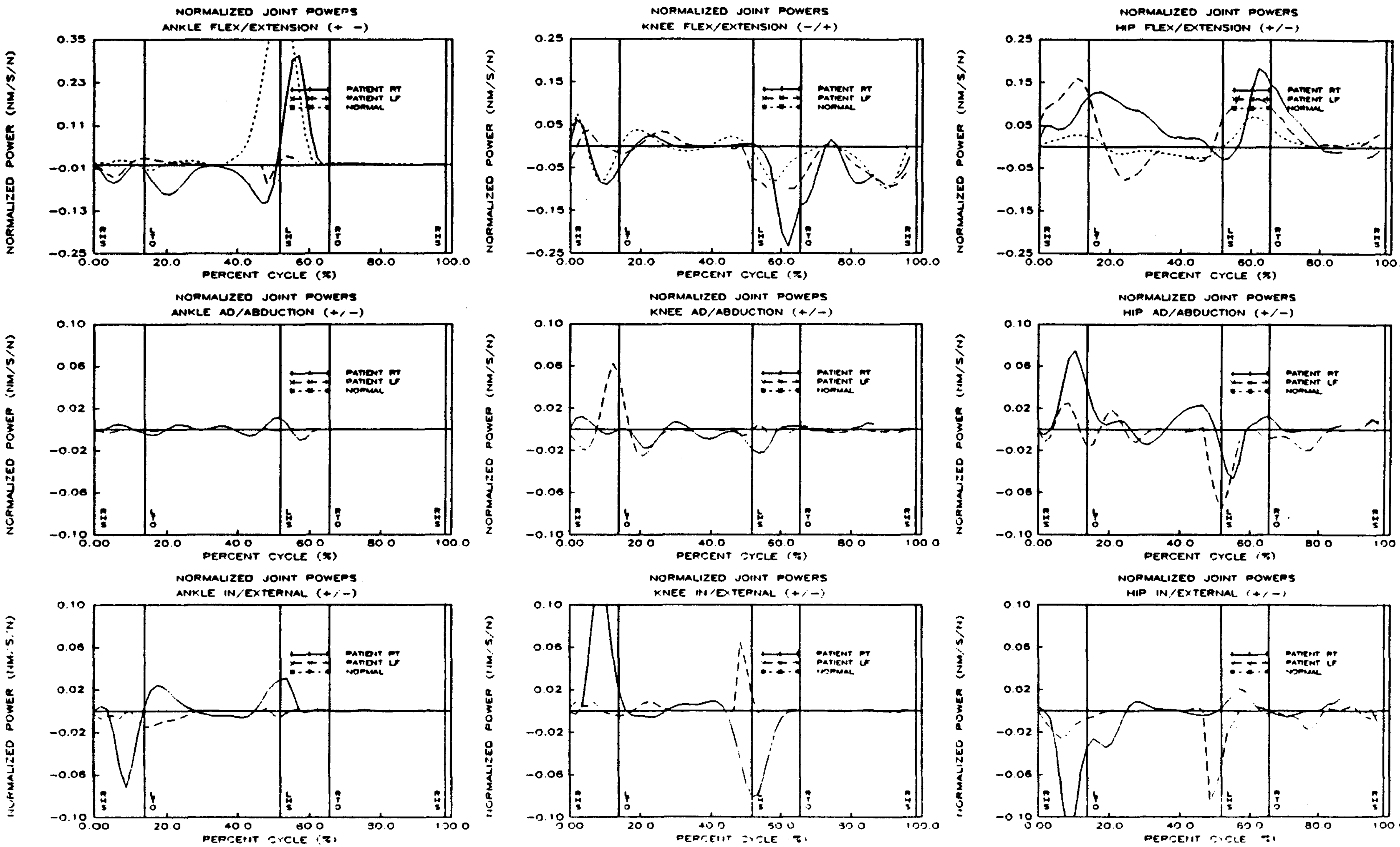
JOINT POWER NORMWTHT : JCS

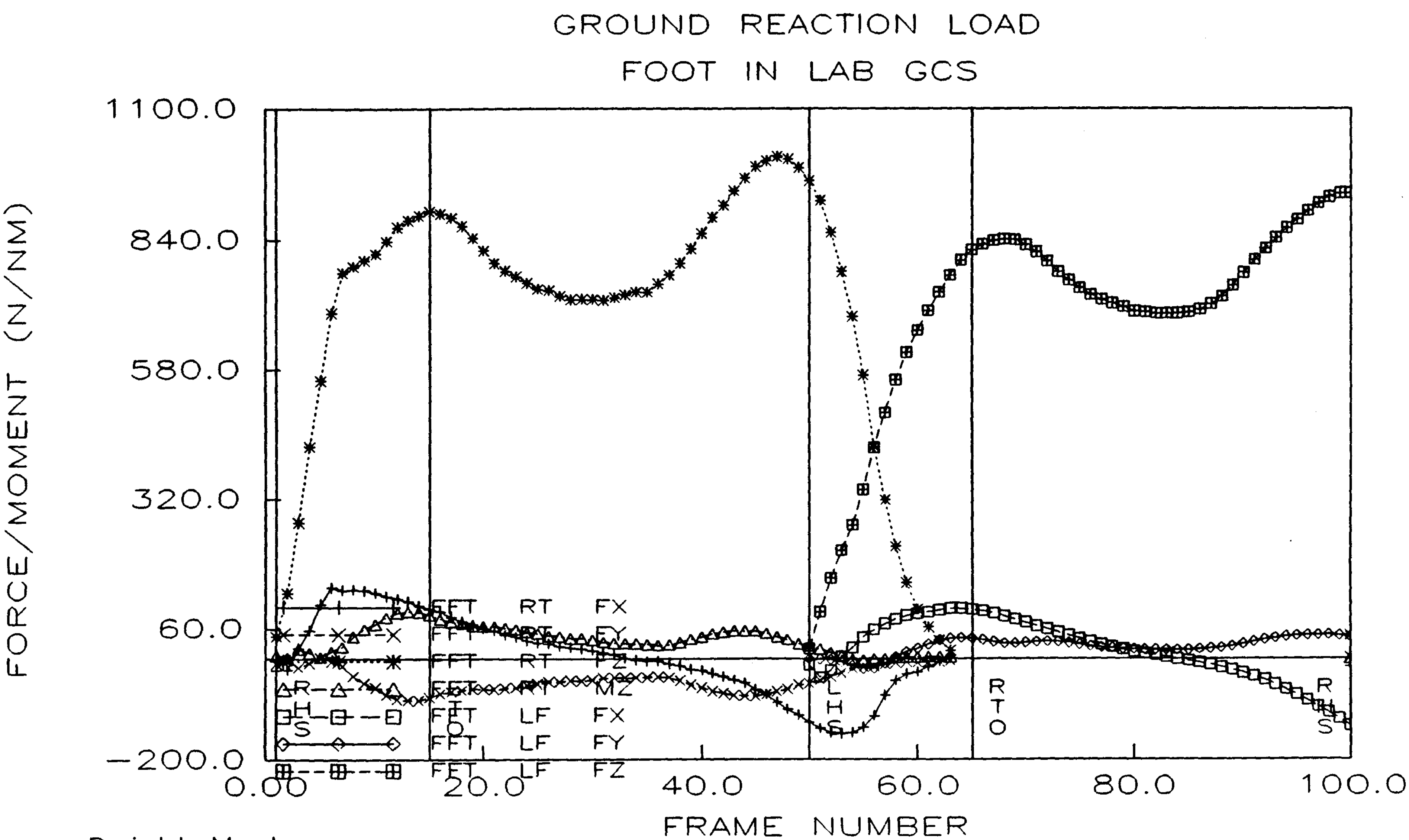


Dwight Meglan

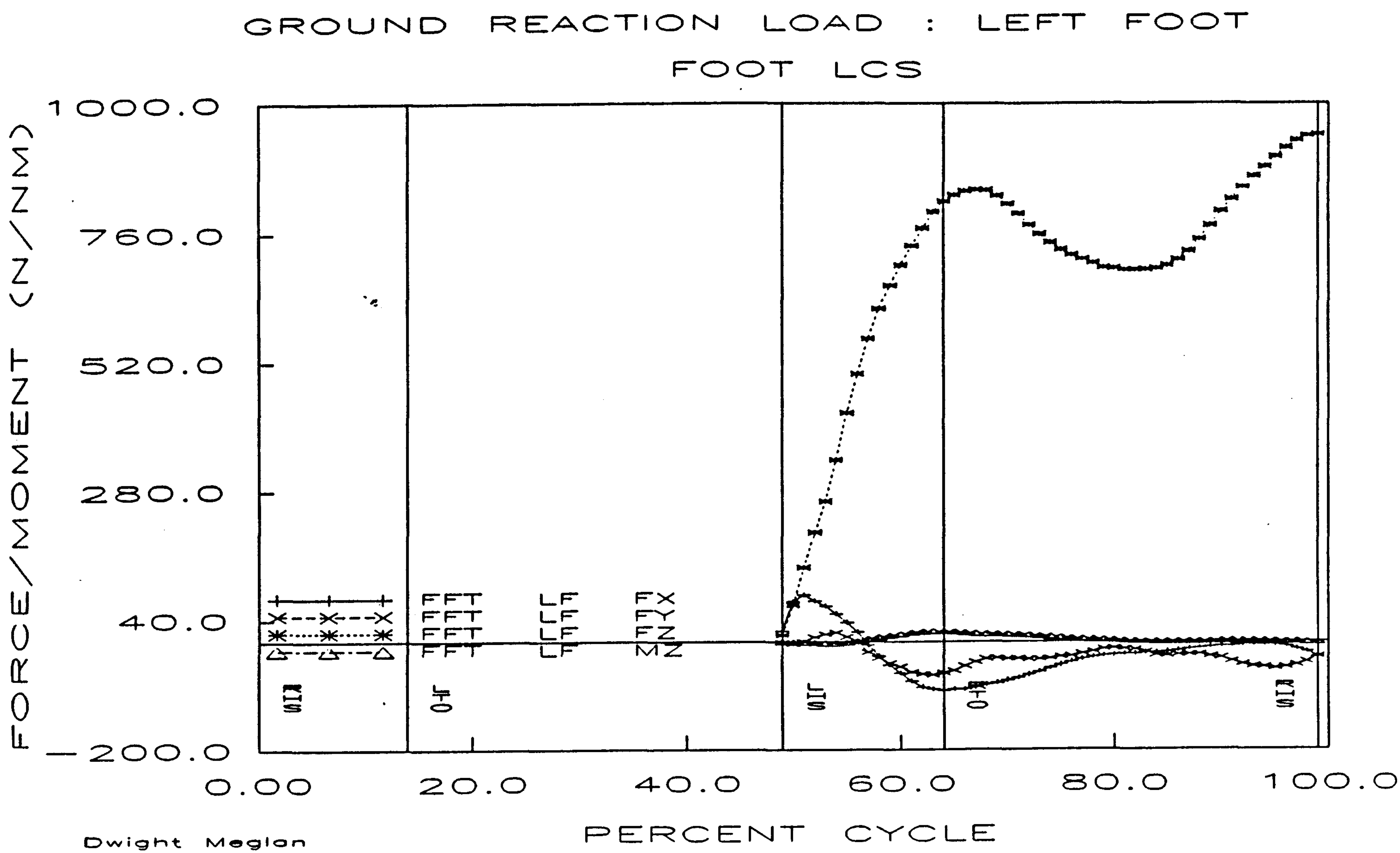

PELVIS ANT/POST TILT ANGLE
HIP FLEXION ANGLE
KNEE FLEXION ANGLE
ANKLE FLEXION ANGLE
FOOT/LAB FLEXION ANGLE
PELVIS RIGHT/LEFT OBLIQUITY ANGLE
HIP AB/ADDUCTION ANGLE
KNEE AB/ADDUCTION ANGLE
ANKLE AB/ADDUCTION ANGLE
FOOT/LAB PRO/SUPINATION ANGLE
PELVIS RIGHT/LEFT FORWARD ROTATION
HIP IN/EXTERNAL ROTATION ANGLE
KNEE IN/EXTERNAL ROTATION ANGLE
ANKLE IN/EXTERNAL ROTATION ANGLE
FOOT/LAB TOEIN/OUT ANGLE
ANGLE (DEG)
PERCENT CYCLE (%)


NORMALIZED JOINT MOMENTS
ANKLE FLEX/EXTENSION (+/-)
KNEE FLEX/EXTENSION (-/+)
HIP FLEX/EXTENSION (+/-)
SUPPORT FLEX/EXTENSION (+/-)
ANKLE AD/ABDUCTION (+/-)
KNEE AD/ABDUCTION (+/-)
HIP AD/ABDUCTION (+/-)
SUPPORT AD/ABDUCTION (+/-)
ANKLE IN/EXTERNAL (+/-)
KNEE IN/EXTERNAL (+/-)
HIP IN/EXTERNAL (+/-)
SUPPORT IN/EXTERNAL (+/-)
NORMALIZED MOMENT (NM/N)
PERCENT CYCLE (%)
PATIENT RT
PATIENT LT
NORMAL


NORMALIZED JOINT POWERS
ANKLE FLEX/EXTENSION (+ −)
NORMALIZED JOINT POWERS
KNEE FLEX/EXTENSION (−/+)
NORMALIZED JOINT POWERS
HIP FLEX/EXTENSION (+/−)
NORMALIZED JOINT POWERS
ANKLE AD/ABDUCTION (+/−)
NORMALIZED JOINT POWERS
KNEE AD/ABDUCTION (+/−)
NORMALIZED JOINT POWERS
HIP AD/ABDUCTION (+/−)
NORMALIZED JOINT POWERS
ANKLE IN/EXTERNAL (+/−)
NORMALIZED JOINT POWERS
KNEE IN/EXTERNAL (+/−)
NORMALIZED JOINT POWERS
HIP IN/EXTERNAL (+/−)
NORMALIZED POWER (NM/S/N)
PERCENT CYCLE (%)
PATIENT RT
PATIENT LF
NORMAL
RHS
LTO
LHS
RTO
RHS

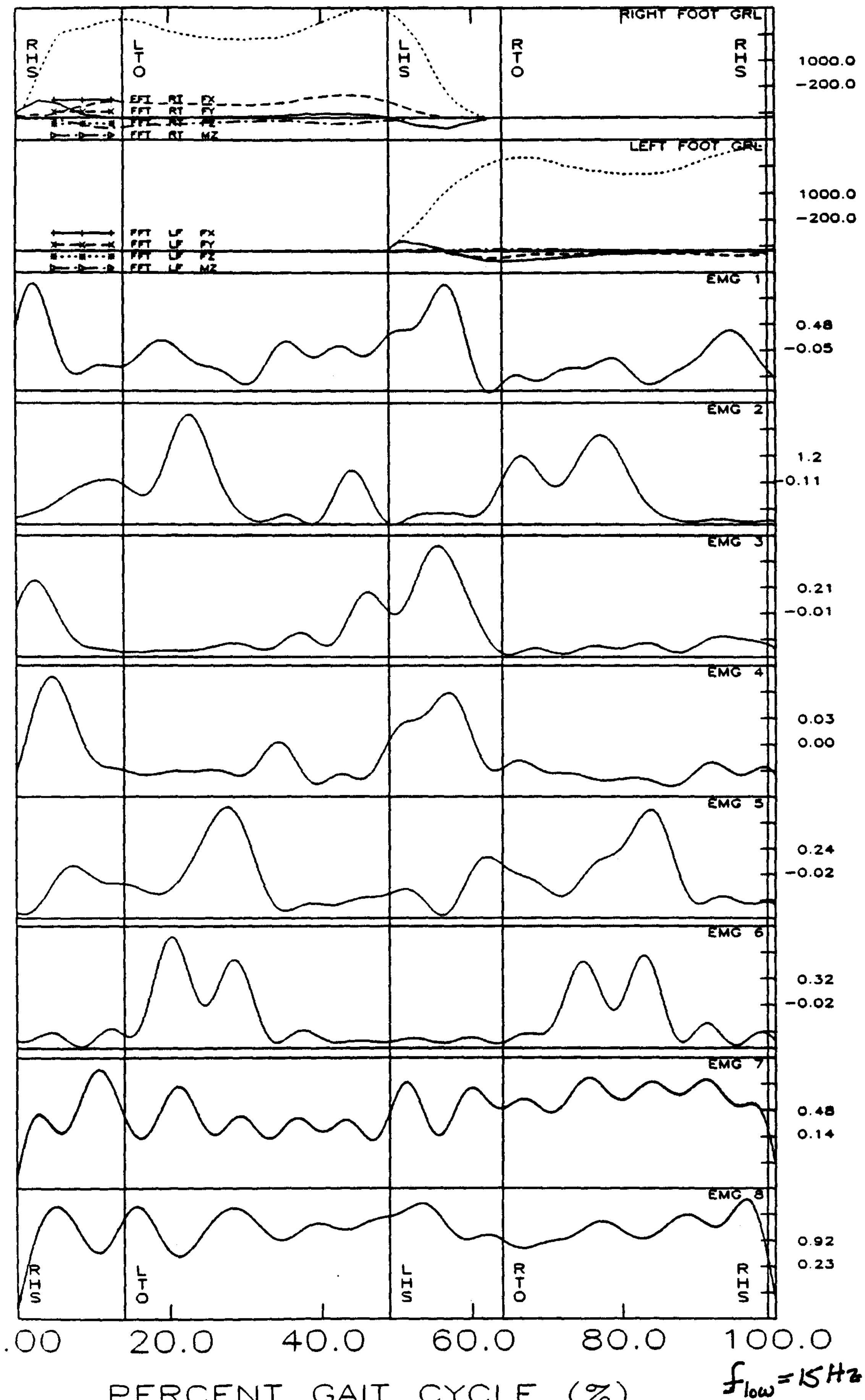


Dwight Meglan



GROUND REACTION LOAD : LEFT FOOT
FOOT LCS
FORCE/MOMENT (N/NM)
1000.0
760.0
520.0
280.0
40.0
-200.0
0.00
20.0
40.0
60.0
80.0
100.0
PERCENT CYCLE
FFT LF FX
FFT LF FY
FFT LF FZ
FFT LF MZ
Dwight Meglan

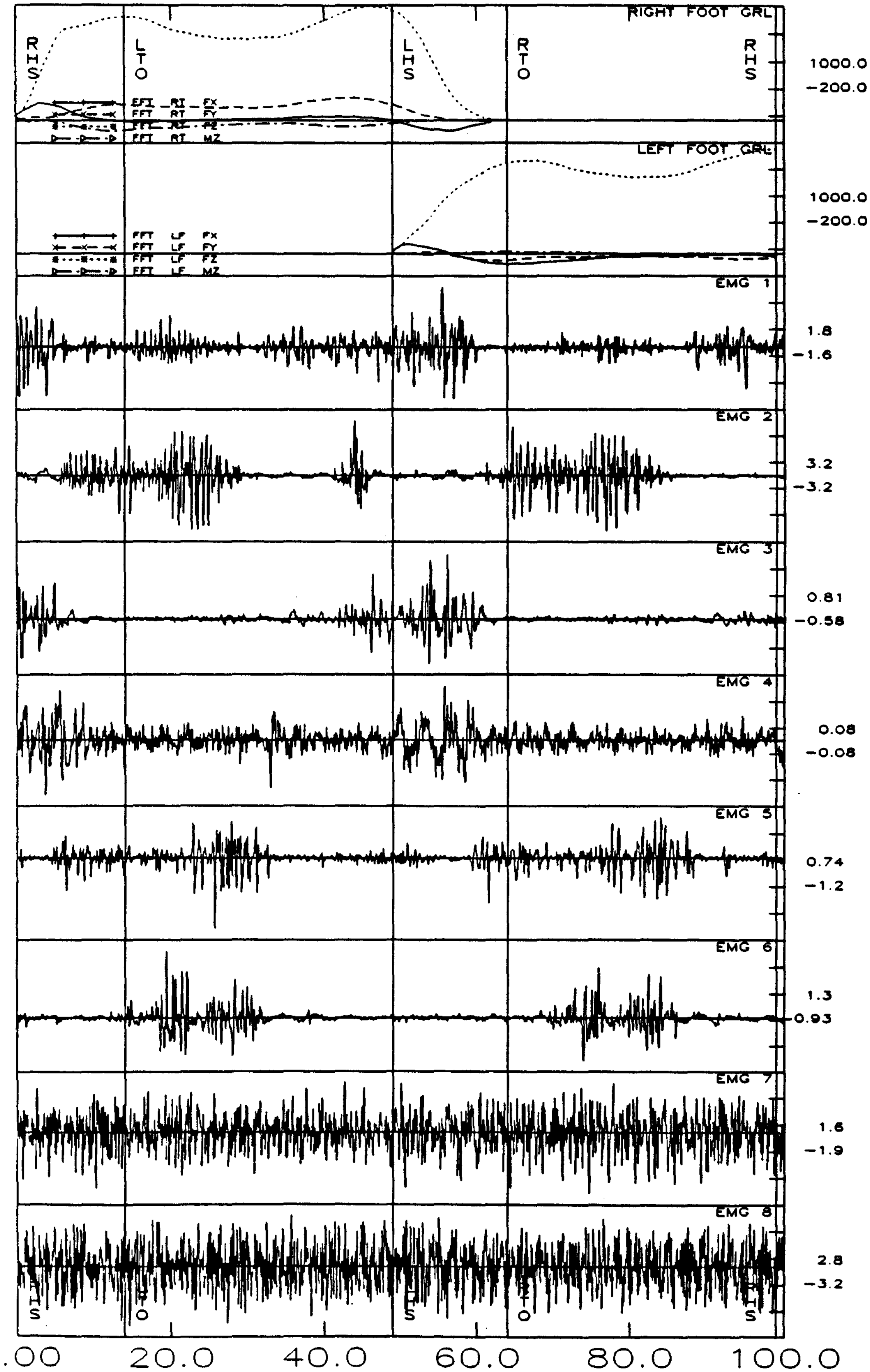
EMG AND FORCE PLATE DATA



Dwight Meglan

f_{low} = 15 Hz f_{High} = 250 Hz
Rectified 6 Hz low pass

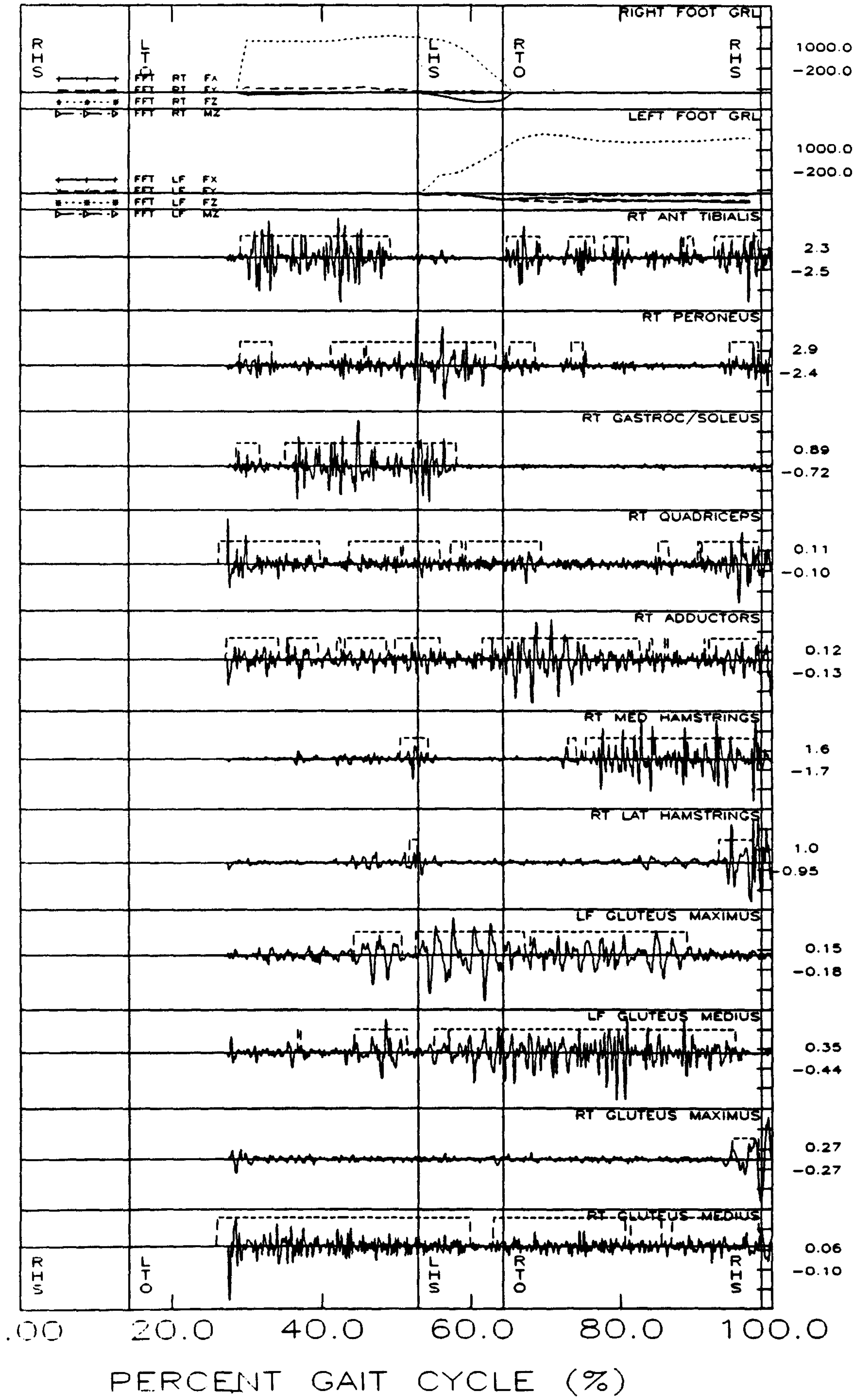
EMG AND FORCE PLATE DATA

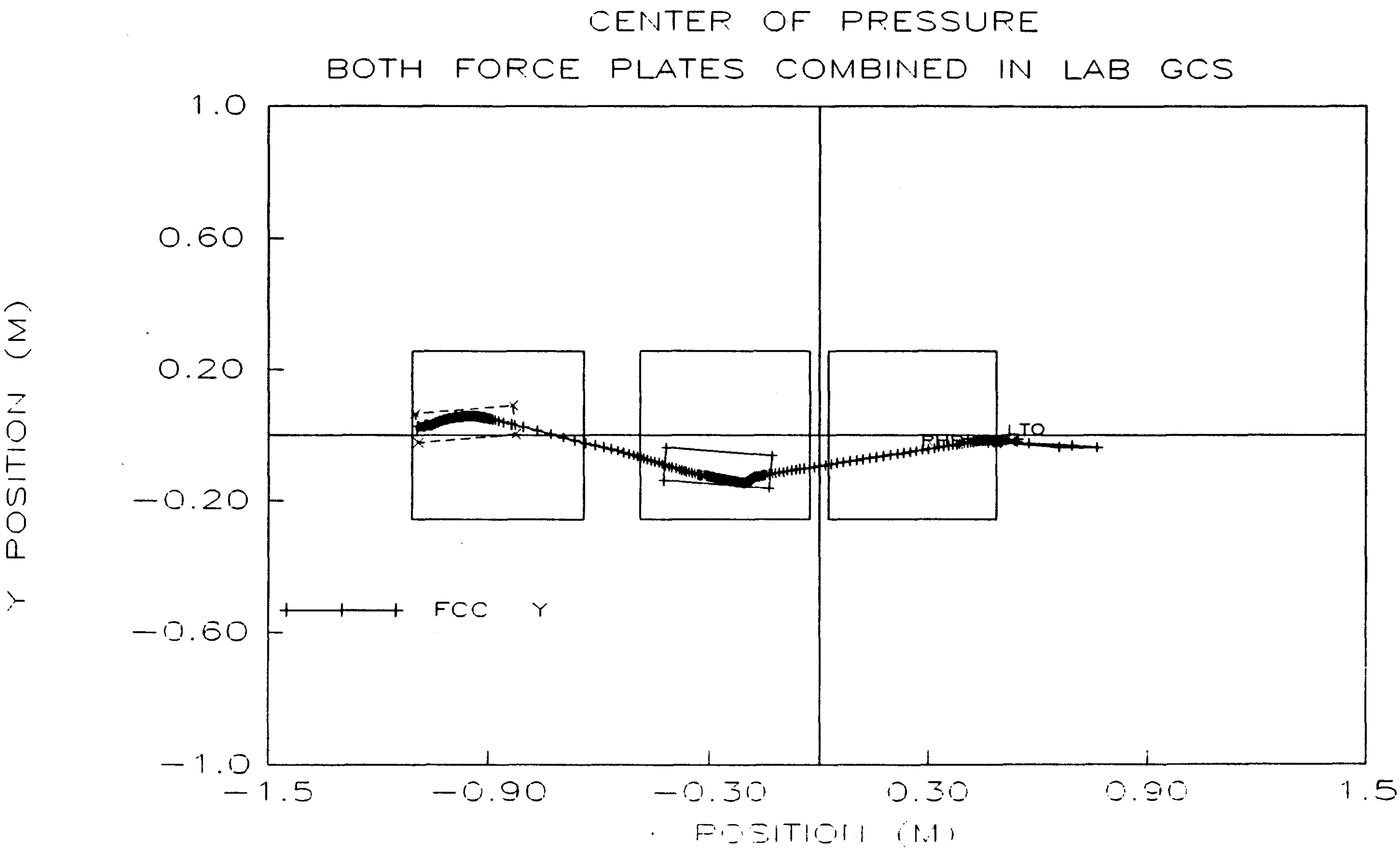


Dwight Meglan

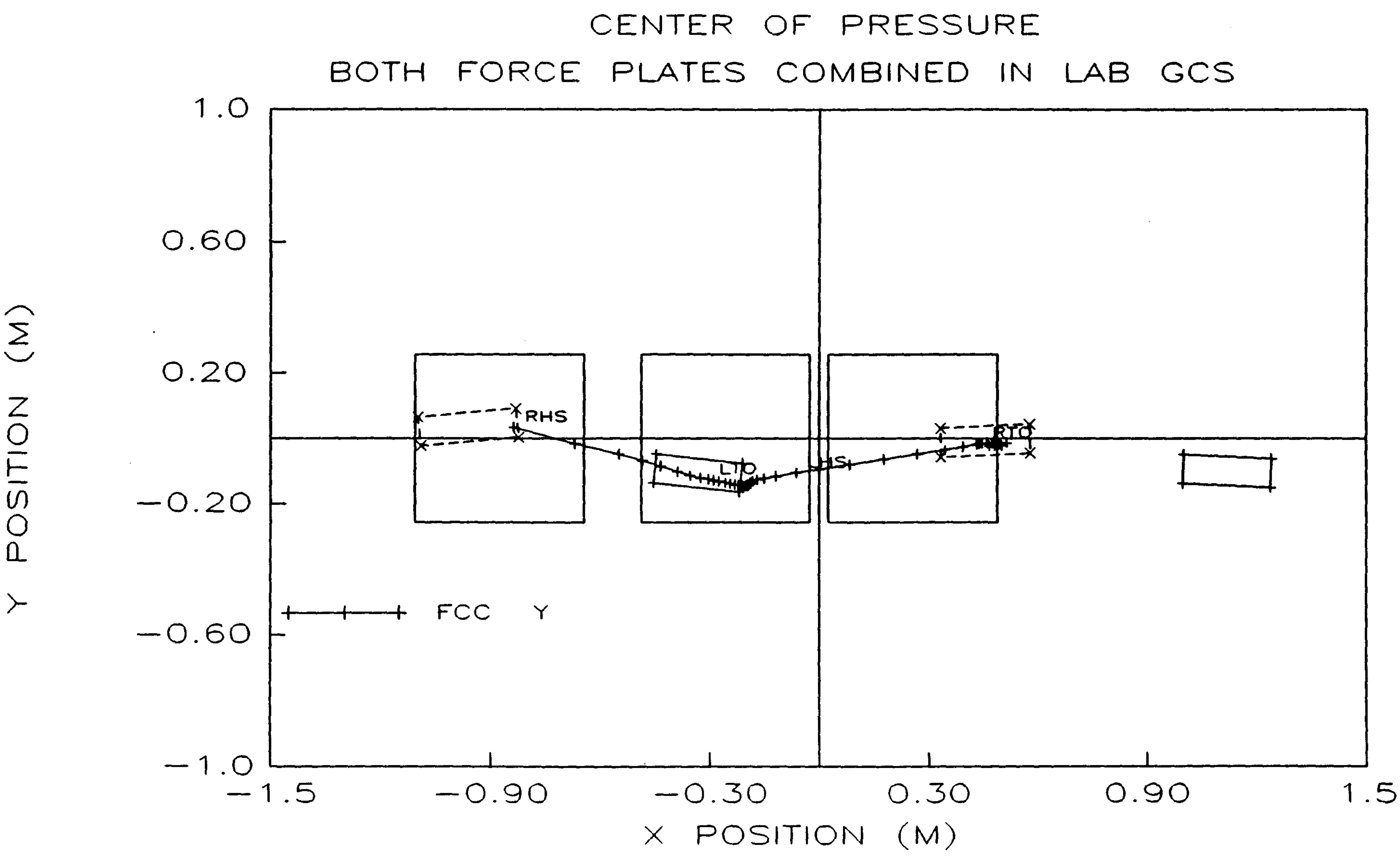
$f_{low} = 15 Hz$ $f_{high} = 250 Hz$

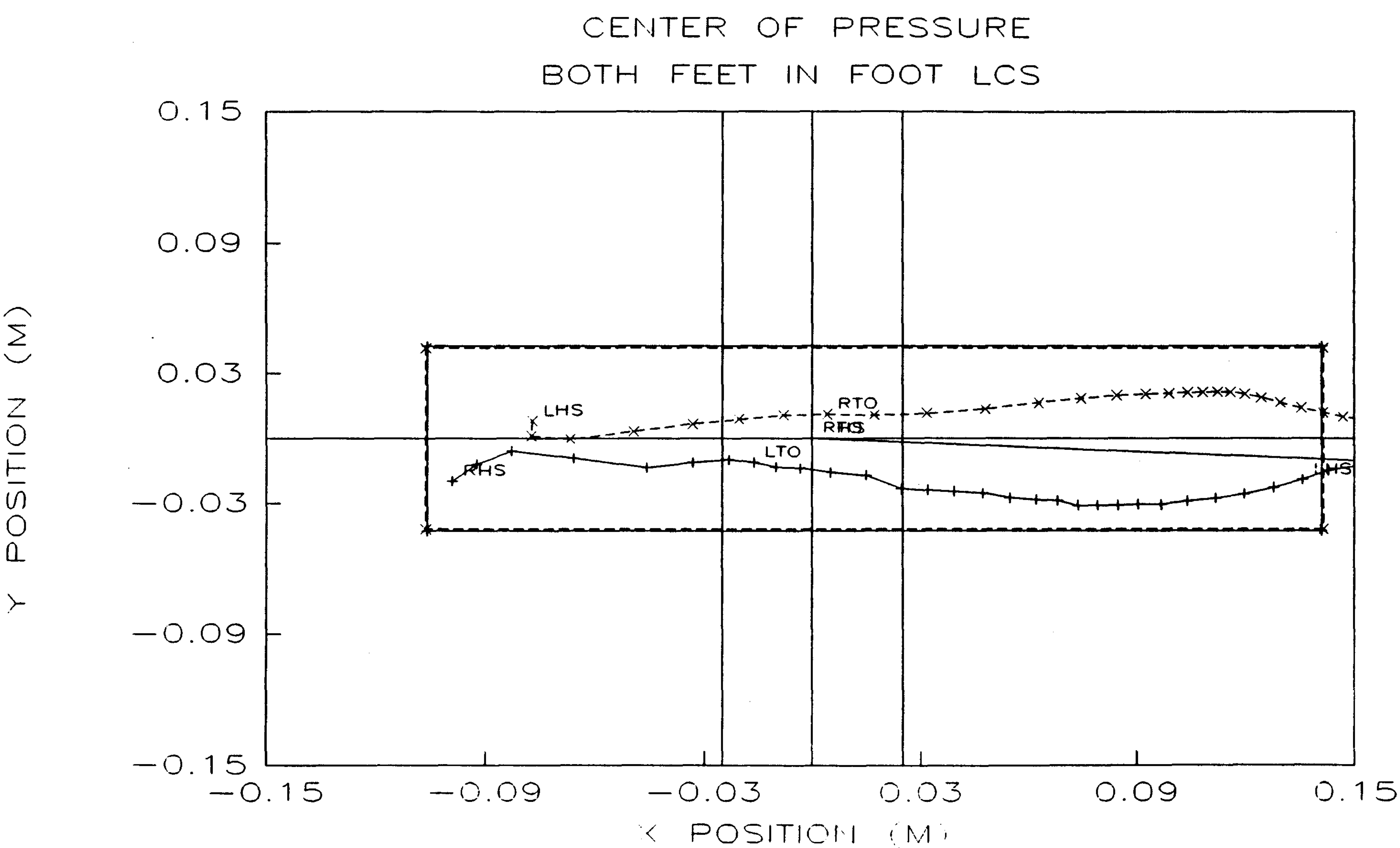
EMG AND FORCE PLATE DATA





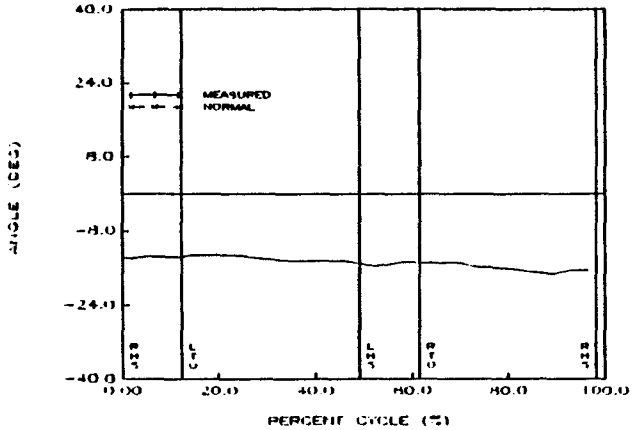
CENTER OF PRESSURE
BOTH FORCE PLATES COMBINED IN LAB GCS
1.0
0.60
0.20
−0.20
−0.60
−1.0
−1.5
−0.90
−0.30
0.30
0.90
1.5
Y POSITION (M)
POSITION (M)
FCC Y
LTO


CENTER OF PRESSURE
BOTH FORCE PLATES COMBINED IN LAB GCS
1.0
0.60
0.20
−0.20
−0.60
−1.0
−1.5
−0.90
−0.30
0.30
0.90
1.5
Y POSITION (M)
X POSITION (M)
RHS
LTO
LHS
RTO
FCC Y


CENTER OF PRESSURE
BOTH FEET IN FOOT LCS
0.15
0.09
0.03
−0.03
−0.09
−0.15
Y POSITION (M)
LHS
RTO
LTO
RHS
−0.15
−0.09
−0.03
0.03
0.09
0.15
X POSITION (M)

PELVIS

AN/POSTERIOR TILT



PUT A COMMENT OR SOMETHING HERE

PUT ANOTHER COMMENT HERE

Gait Distance/Time Parameters

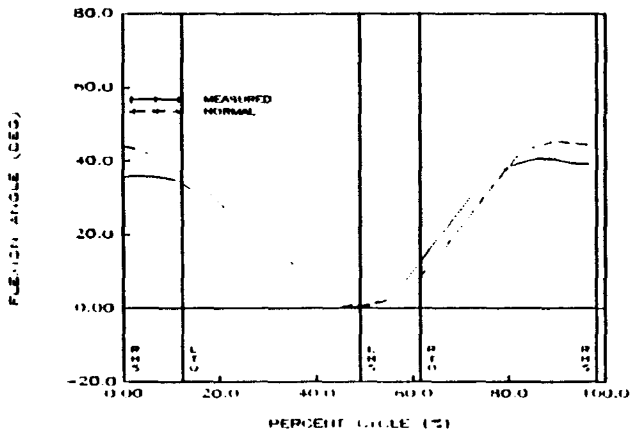
	Meas.(ft)	Normal(ft)	% Normal
Velocity	4.547	4.760	95.521
Stride	5.256	5.144	102.203
Rt Step	2.949	2.572	114.714
Lf Step	2.306	2.572	89.693
Step Width	0.390	0.259	151.091
Cadence	105.263	111.000	94.831

CATH

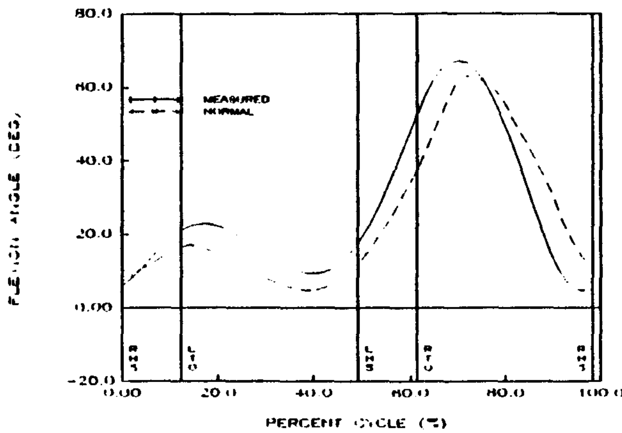
Gait Timing Parameters

	Time(s)	Cycle(%)	Normal
Rt TLST	0.759	66.666	62.000
Lf TLST	0.759	66.666	62.000
Rt SLST	0.439	38.596	38.000
Lf SLST	0.439	38.596	38.000
Rt DLS	0.159	14.035	12.000
LF DLS	0.159	14.035	12.000
RHS		0.000	0.000
LTO		14.035	12.000
LHS		50.877	50.000
RTO		63.157	62.000
RHS		100.000	100.000

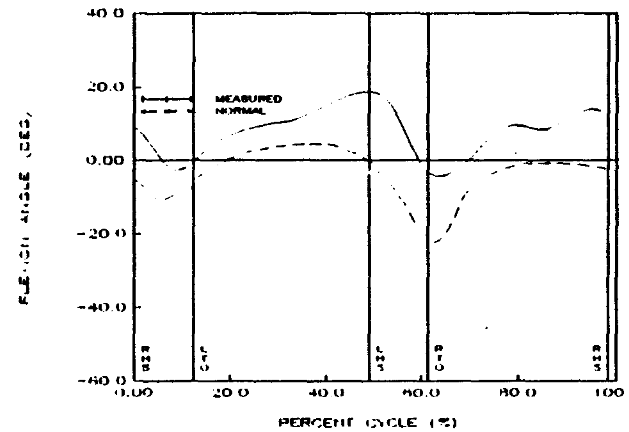
RIGHT HIP FLEXION ANGLE



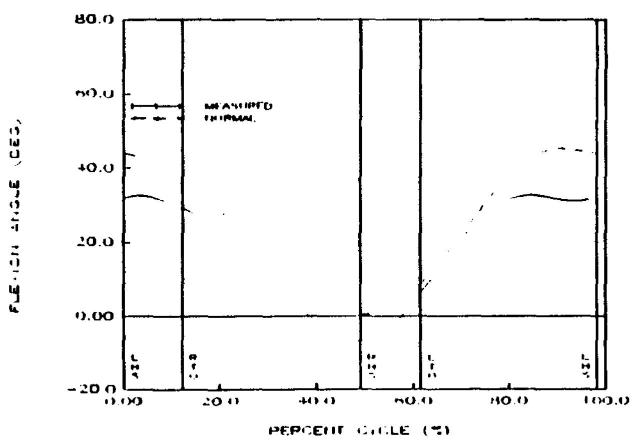
RIGHT KNEE FLEXION ANGLE



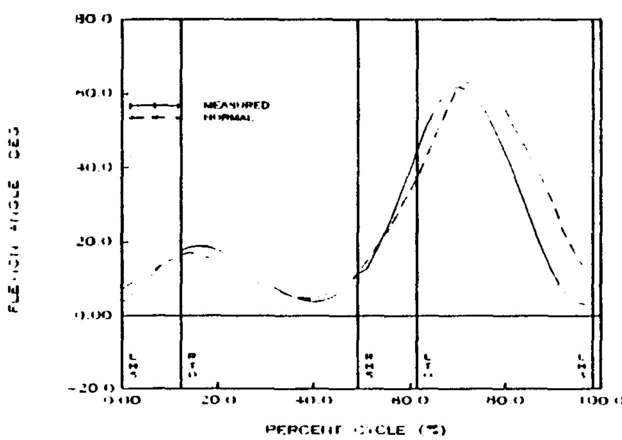
RIGHT ANKLE FLEXION ANGLE



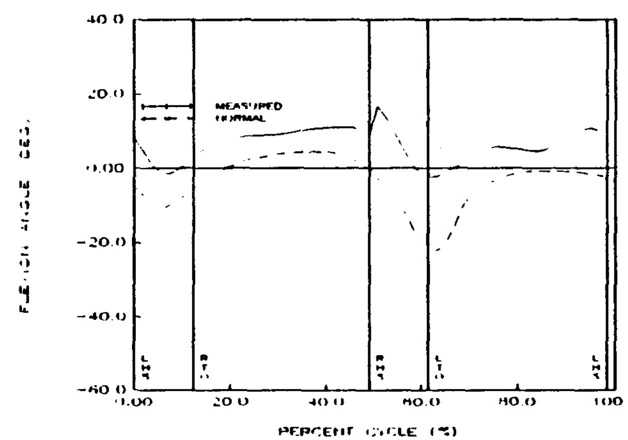
LEFT HIP FLEXION ANGLE

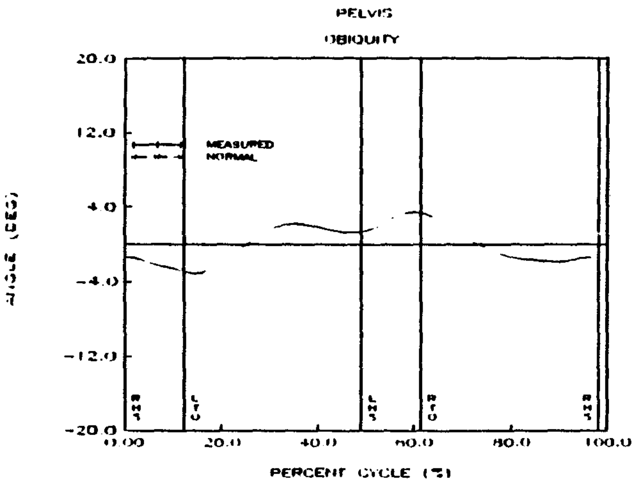


LEFT KNEE FLEXION ANGLE



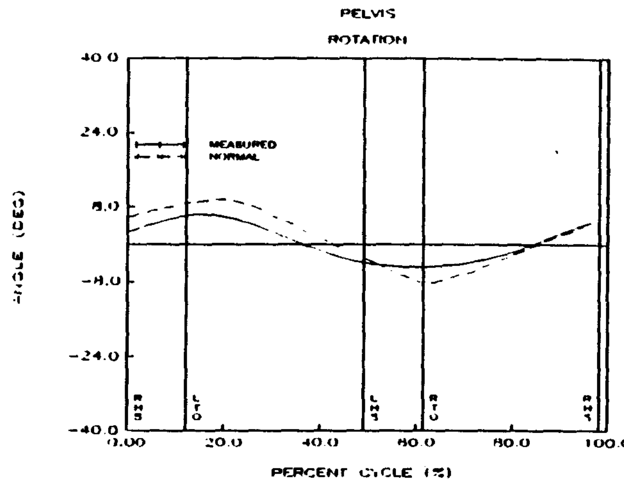
LEFT ANKLE FLEXION ANGLE

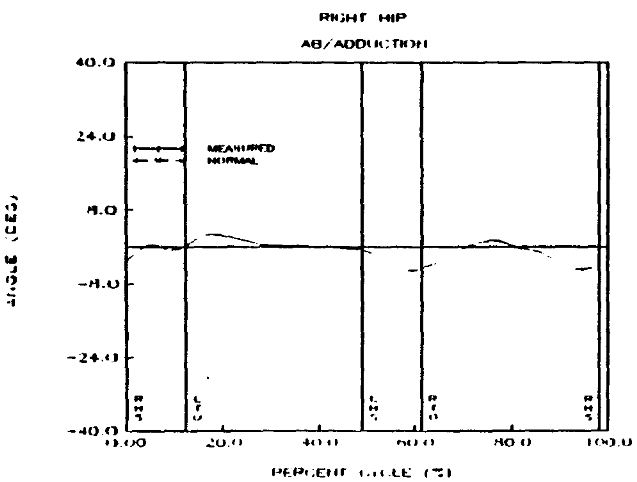


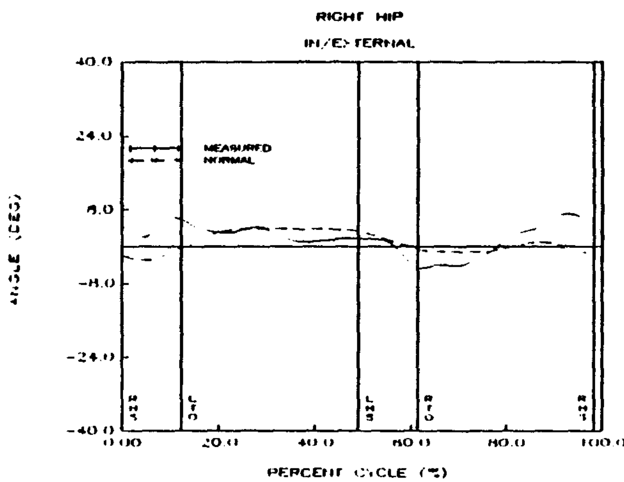


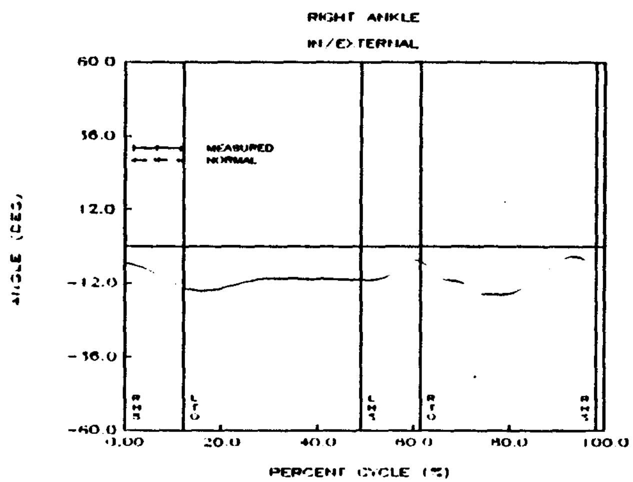
PUT A COMMENT OR SOMETHING OR NOTHING HERE
ANOTHER AVAILABLE LINE

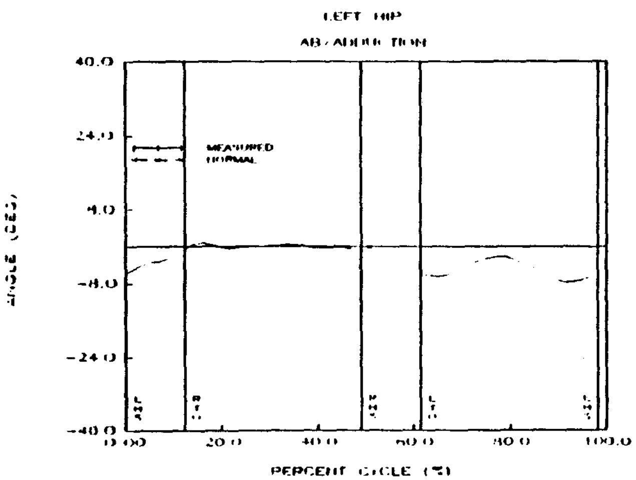
26-AUG-91
CATHY8_JLD AN2

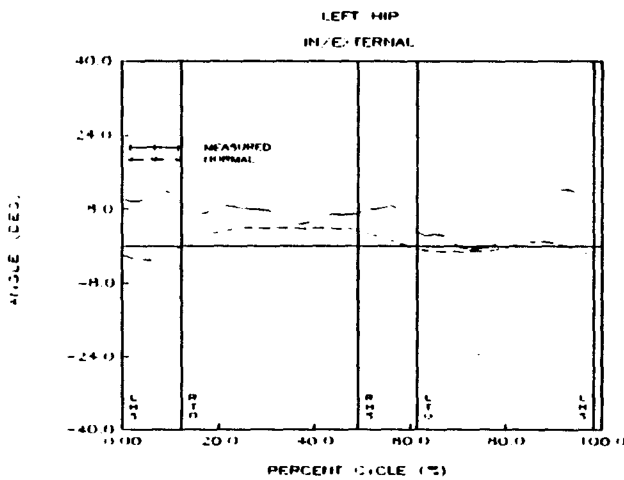


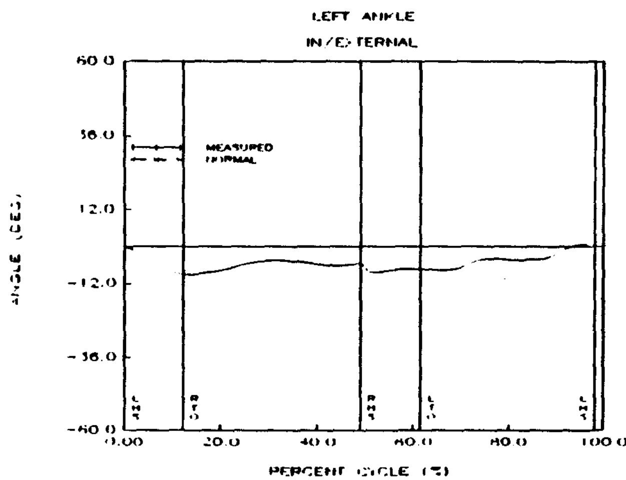












Name : Jeff
Age : 37
Weight : 62.136 kg
Height : 1.854 m
Comment:

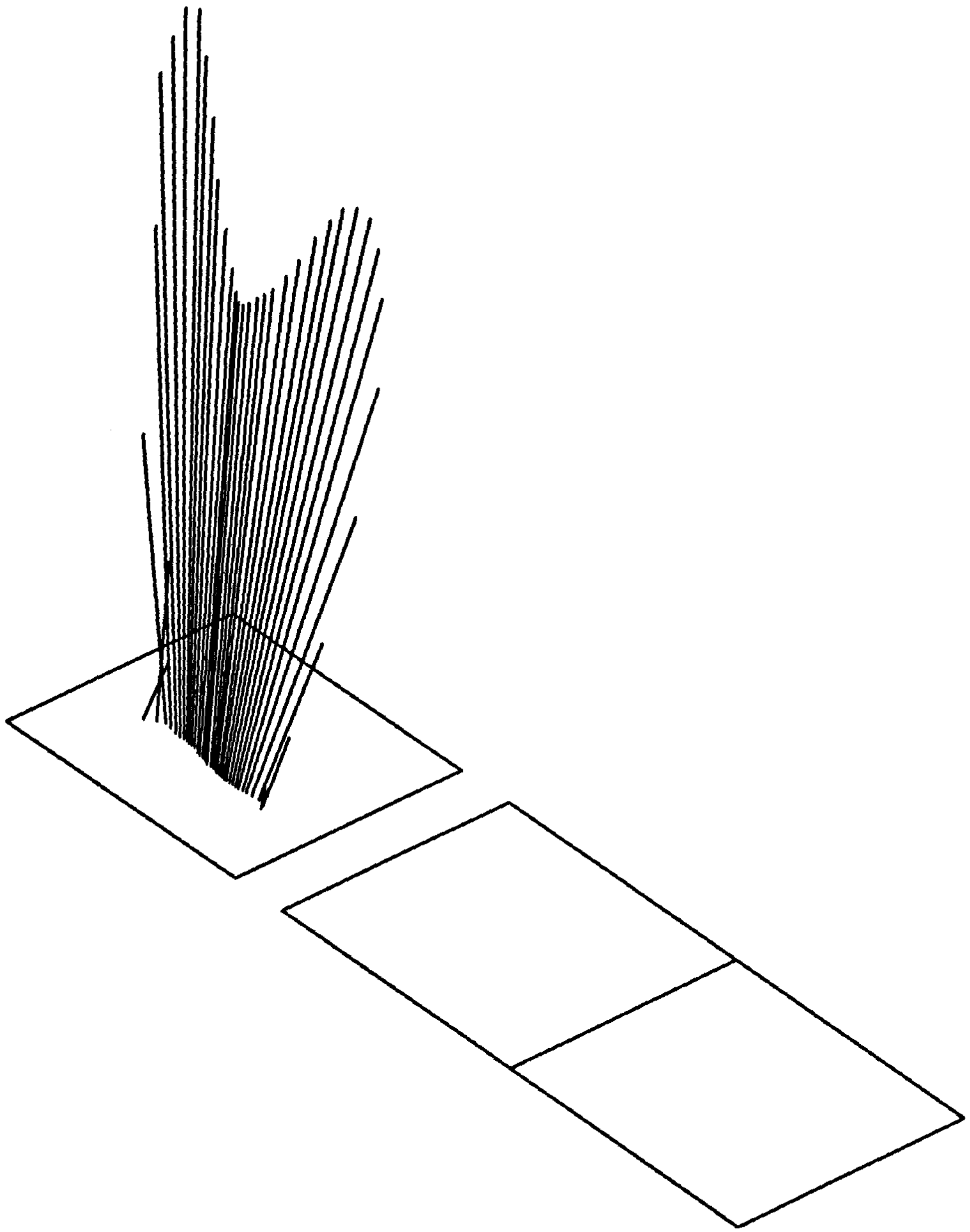
Cycle event frame #'s

Right heel strike : 1 57
Left heel strike : 29
Right toe off : 36
Left toe off : 8

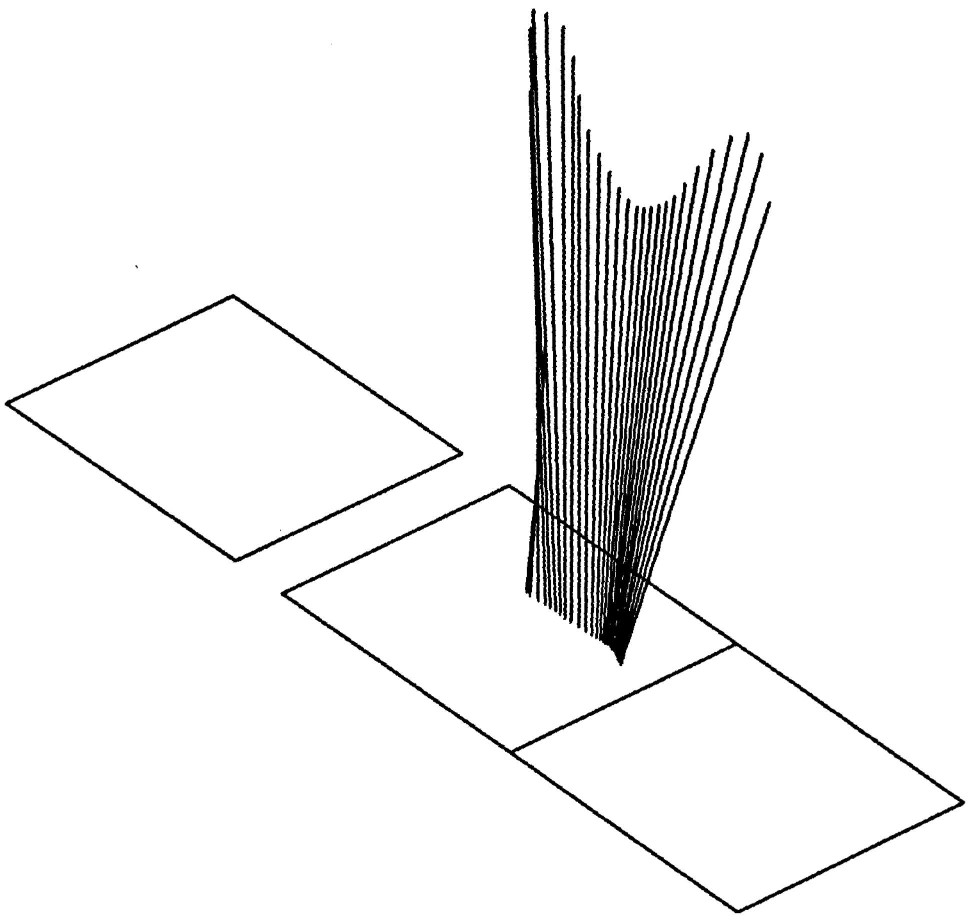
First motion frame absolute frame # : 1

		meters	feet	inches	Normal(m)	% Normal
Velocity	:	1.386	4.547	54.567	1.451	95.521
Stride	:	1.603	5.258	63.092	1.568	102.204
R Step	:	0.899	2.951	35.407	0.784	114.714
L Step	:	0.703	2.307	27.684	0.784	89.693
Step Width	:	0.119	0.392	4.699	0.079	151.092
Cadence	:	105.263 step/min			111.000	94.832

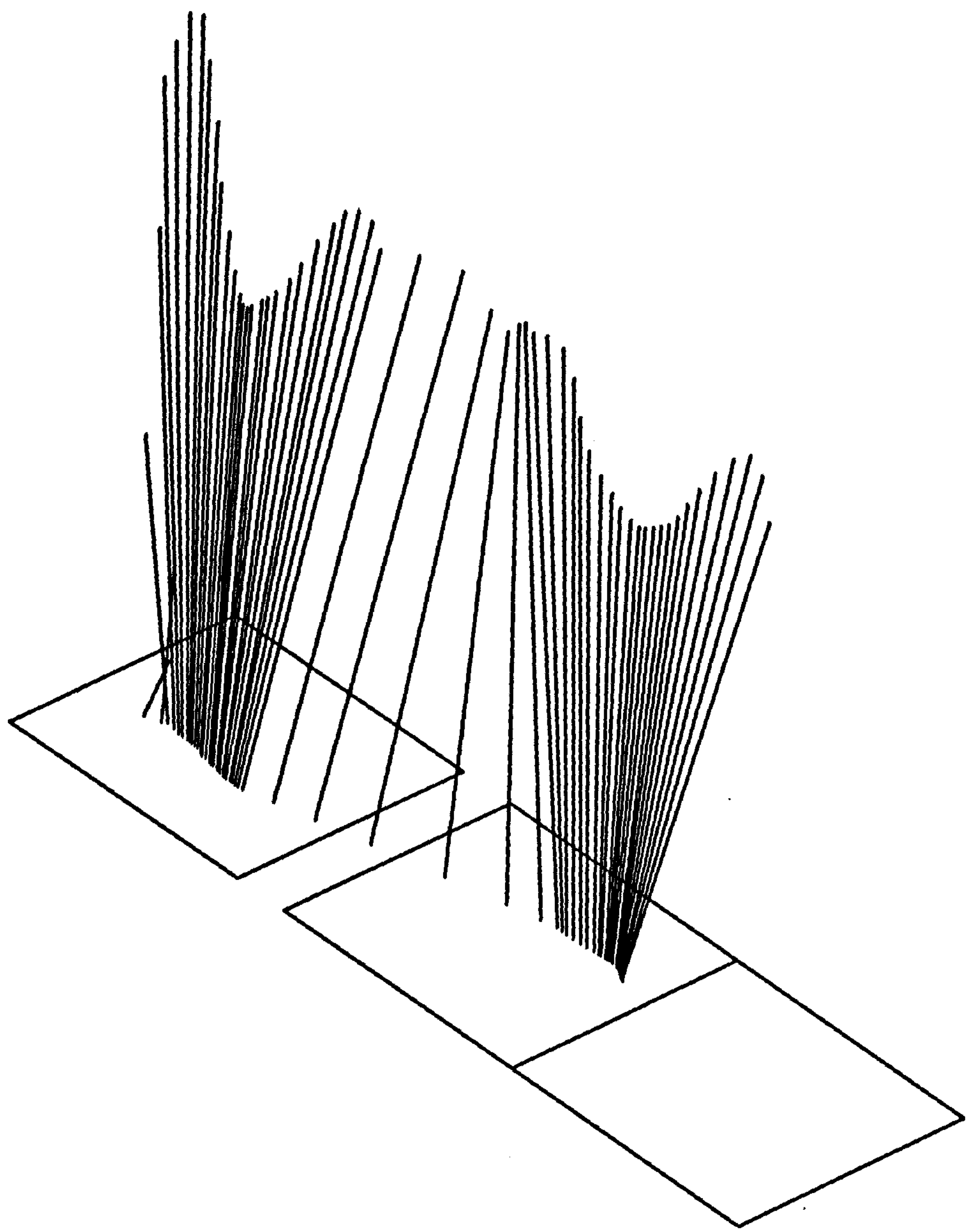
	Time (s)	Percent Cycle	Normal
R TLST	0.600	66.632	62.000
L TLST	0.600	66.632	62.000
R SLST	0.440	38.596	38.000
L SLST	0.440	38.596	38.000
R DLS	0.160	14.035	12.000
L DLS	0.160	14.035	12.000
RHS		0.000	0.000
LTO		14.035	12.000
LHS		50.877	50.000
RTO		63.158	62.000
RHS		100.000	100.000



File : TR3_DIR:CATHY.TR3;8
Comment :
Plot Date : 17-AUG-92



s : TR3_DIR:CATHY.TR3;8
mment :
t Date : 17-AUG-92



File : TR3_DIR:CATHY.TR3;8
Comment :
Plot Date : 17-AUG-92

ANZ Command Summary

[]	indicates default value
**	indicates default active option
(#)	indicates number required with option

Section	Command	Option(s)
AVeraGe	POSition	
	ANGle	
	REFerence	/FILe=filename file with reference position data
CYCle	EVEent	
	STAtistics	/use ANKle marker to calculate step stats [no]
		/use HEEl marker to calculate step stats [no]
		/use TOE marker to calculate step stats [no]
EMG	NAMe	
	FILter	/apply hanning WINdow to data **
		/No WiNdowing of data
		/LOW frequency cutoff (# hz)
		/Low frequency cutoff SLope (# %/hz)
		/HIGh frequency cutoff (# hz)
		/High frequency cutoff SLope (# %/hz)
		/NOTch frequency center (# hz)
		/Notch WIdth (# hz)
		/Notch frequency side SLope (# %/hz)
	RECtify	
	INTegrate	/WINdow width (# msec) [50.0 msec]
	MoveWinAvg	/WINdow width (# msec) [50.0 msec]
		/use COSine taper on window [no]
	THReshold	/PERcent (# %) [20.0 %]
FoRCe	CONdition	/NO Zeroing of force plate data [no]
		/variation allowed in IDLe region (# % fz) [5.0 %]
		/minimum # of FRaMes in idle region (#) [10]
		/FZ Minimum plate active threshold (# N) [10]
		/fz in IDle region Maximum level (# N) [10]

ALIgn /manually assign Right FooT plates (#) [none]
/manually assign Left FooT plates (#) [none]
/frame of Heel Strike on plate 1 (#) [none]
/frame of Heel Strike on plate 2 (#) [none]
/frame of Heel Strike on plate 3 (#) [none]
/first force plate hit by SECond heelstrike [no]
/FZ Minimum active theshold level (# N) [5.0]
/Foot Strike Threshold (# % cycle) [10.0]
/change in fz TRIgger for heelstrike (# % fz) [10.0]
/CHAnge in fz level detection of heelstrike **
/ABSolute value of fz level detection of heelstrike [no]
/MZ moment Zero on plate 1 [no]
/use 2ND foot heelstrike & GRL data to find frame offset [no]

CLIp force/motion data

Center of PRessure

Local Center Pressure

WREnch

INTegral /estimate Body Center of Mass path [no]
/set inital bcm VeLocity of Zero [no]

REConstructed grl

Pitch Moment Arm

Pitch Wrench Arm

Center of Pressure Velocity

FILter /apply hanning WINdow to data **
/No WiNdowing of data
/LOW frequency cutoff (# hz)
/Low frequency cutoff SLope (# %/hz)
/HIGh frequency cutoff (# hz)
/High frequency cutoff SLope (# %/hz)
/NOTch frequency center (# hz)
/Notch WIdth (# hz)
/Notch frequency side SLope (# %/hz)

HIStory

JoiNT Jcs ANgle /use REFerence joint angles [no]
/modify ankle Ankle FLexion (# deg) [no, -6.0]
/No Ankle Flexion modification **
/use SHoulder angle Polar definition [no]
/Extrapolation THreshold using REF (# deg) [10.0]
/FILe=filename file with reference angle data
/INTegrate joint angle velocities to get joint angles

Jcs Vel Accel /use SIMple differentiation of JCS angles [no]
/GeneralizedCrossValidation **
/DynamicProgrammingFilter [no]
/NOIse added % of signal range (# %) [0.5]
/NO Noise added to signal for dpf
/VARiability threshold for no smoothing (# %) [1.0]

Euler ANgle /modify ankle Ankle FLexion (# deg) [no, -6.0]
/No Ankle Flexion modification **
/use SHoulder angle Polar definition [no]

Euler Vel Acc /use SIMple differentiation of JCS angles [no]
/GeneralizedCrossValidation **
/DynamicProgrammingFilter [no]
/NOIse added % of signal range (# %) [0.5]
/NO Noise added to signal for dpf
/VARiability threshold for no smoothing (# %) [1.0]

LOAd /use FULl body calculation scheme [no]
/INTerpolate bad frames [no]
/No INterpolation [no]
/interpolation POLynomial order (#) [3]
/OVErride the bad GRL indicators near a gait event **
/NOVerride of bad GRL indicators

WREnch

POWer

TRaNslation

ScrewANgle

MomentARm

WrenchARm

FILter /apply hanning WINdow to data **
/No WiNdowing of data
/LOW frequency cutoff (# hz)
/Low frequency cutoff SLope (# %/hz)
/HIGh frequency cutoff (# hz)
/High frequency cutoff SLope (# %/hz)
/NOTch frequency center (# hz)
/Notch WIdth (# hz)
/Notch frequency side SLope (# %/hz)

Filterable joint quantities: JAN JVA LOA POW

EXternal Moment

WoRK

LoadANgle

CENter /Finite SCrew axis [no]
/Instataneous SCrew axis **
/intersect screw with PLaNe= filename **
/use SPHerical motion assumption [no]

	AverageCeNter	/limit centers to CLIpping volume= filename
		/use SEQuential motion frames for joint centers **
		/use PERmutations of all available motion frames [no]
MaRKer	INTerpolate	/POLynomial order [3]
		/EXTrapolation polynomial order [3]
	ADJust	
	SMOoth	/VelocityAndAcceleration [no]
		/GeneralizedCrossValidation **
		/DynamicProgrammingFilter [no]
		/NOIse added % of signal range (#) [0.5 %]
		/NO Noise added to signal for dpf
		/VARiability threshold for no smoothing (#) [1.0%]
	FILter	/apply hanning WINdow to data **
		/No WiNdowing of data
		/LOW frequency cutoff (# hz)
		/Low frequency cutoff SLope (# %/hz)
		/HIGh frequency cutoff (# hz)
		/High frequency cutoff SLope (# %/hz)
		/NOTch frequency center (# hz)
		/Notch WIdth (# hz)
		/Notch frequency side SLope (# %/hz)
	REGularize	/use gait CYCle for start & end frames [no]
		/STArt frame (#) [1]
		/END frame (#) [number of motion frames]
		/PoiNTs on interval (#) [100]
		/interpolation POLynomial (#) [3]
		/PAD points (#) [0]
OPTion	GENeral	LOGfile
		UPDate display
		NOUpdate
		INTerpolating polynomial order
	FoRCe	number of idle FRaMes
		IDLe region variation
		plate active TRIgger value of fz
		HeelStrike Threshold

READ	/TR3	override file extension check: read TR3 file format
	/TRU	override file extension check: read TRU file format
	/C3D	override file extension check: read C3D file format
	/ANZ	override file extension check: read ANZ file format
	/GL1	override file extension check: read old GLS file format
	/GL2	override file extension check: read new GLS file format
	/FPD	override file extension check: read FPD file format
	/NCZ	file marker data uses new Coord system [no]
	/OGN	force plate data uses old amplifier gains [no]
	/ZER	zero force plate data using zeros in file [no]
	/NOC	don't convert GLS2 force plate data with cal matrix [no]
	/BEG	start frame for reading GLS1 and GLS2 data [1]
	/DRZ	resolution decrease for reading GLS1/GLS2 (#) [no, 10]
	/UPP	data in C3D file is for upper extremity study [no]
	/NOH	upper extremity data does not contain head markers [no]
	/BYT	reverse byte order of dat in ANZ file [no]
	/NOF	don't use the calculation flags stored in the ANZ file [no]
	/NMN	dont't override marker names in file [no]
	/TDP	data in C3D file is for time-distance parameter calculation only [no]

SAVe	/TRU	save marker/angle data in TRU file format
	/GLS	save force plate/EMG data in GLS2 file format
	/ANZ	save all possible data in ANZ file format **
	/NOF	don't save the calculation flags [no]

SEGment	PARameters	/ZeroMassInertia [no]
		/ZeroNonSagittal inertias [no]
		/ZerINertia only [no]
		/STArt frame (#) [1]
		/NUMber of frames (#) [number of motion frames]
		/force use of ADUlt bsp formulas [no]
		/force use of CHIld bsp formulas [no]
	MaRKer diagnostics	/ANGle between markers rather than distance
	POSition	/use anatomic REFerence correction [no]
	VelAndAccel	/use yxz EULer angle derivatives **
		/use rotation MATrix derivatives [no]
		/use old method for ROTation angle derivs [no]
		/SMOoth the rotation matrix components [no]
		/use MaRKer vel/accel [no]
		/use RIGid body correction with MRK [no]
		/generate DIAgnostics with MRK [no]
		/GeneralizedCrossValidation **
		/DynamicProgrammingFilter [no]
		/NOIse added % of signal range (#) [0.5 %]
		/NO Noise added to signal for dpf
		/VARiability threshold for no smoothing (#) [1.0%]
	ENeRgy	

```
FiniteSCrew  /use WALdron's method **
             /use WOLtring's method [no]
             /do ABSsolute calculation **
             /do RELative calculation [no]

InstantSCrew

BodyCenMass      /calculate bcm VELocity also

MOMentum

ANGle     /INTegrate the segment rotational acceleration

FILter       /apply hanning WINdow to data **
             /No WiNdowing of data
             /LOW frequency cutoff (# hz)
             /Low frequency cutoff SLope (# %/hz)
             /HIGh frequency cutoff (# hz)
             /High frequency cutoff SLope (# %/hz)
             /NOTch frequency center (# hz)
             /Notch WIdth (# hz)
             /Notch frequency side SLope (# %/hz)

   Filterable segment quantities:  TVL   RVL   POS   VAA
                                      TAC   RAC


STAtus  GENeral
        PATient
        MaRKer
        SEGment   <none>
                  INErtia
        FoRCe     <none>
                  AVerGe
        JoiNT
        AVeraGe   <none>
                  POSition
                  ANGle  JCS
                         EULer
        CYCle
        EMG
        LIMit
```

Data Export Commands

```
EXPort      AVeraGe       POSition/orientation
/IGOr                         Jcs ANgle
/NOHeader                     anatomic REFerence correction
/INC=#

EXPort      gait CYCle

EXPort      EMG

EXPort      FoRCe      GroundReactionLoad { plt #, comb (/ANI), ft ([/GCS],/LCS) }
                                              (/NWT,/NWH,/NWL)
                                              /LBS

                          CenterPRessure       { plt #, comb, ft }

                          LocalCenterPressure

                          WREnch          { plt #, comb, ft }

                          INtegral1st   { plt #, comb, ft }

                          INtegral2nd  { plt #, comb, ft }

                          StrikeIndex

                          PitchMomentArm          (/GCS,[/PRO],/DIS)

                          PitchWrenchArm          (/GCS,[/PRO],/DIS)

                          CenterPressureVelocity

EXPort      HIStory

EXPort      JoiNT      JcsANgle

                          JcsVelocity&Acceleration        /NOR

                          EulerANgle

                          EulerVelocity&Acceleration      /NOR

                          LOAd        (/NWT,/NWH,/NWL)
                                      /LBS
                                      ([/JCS],/GCS,/PRO,/DIS)

                          LegLoaD     (/NWT,/NWH,/NWL)
                                      /LBS
                                      ([/JCS],/GCS)

                          WREnch

                          POWer  (/NWT,/NWH,/NWL)
                                      /LBS
```

```
                                LegPoWer    (/NWT,/NWH,/NWL)
                                            /LBS

                                FiniteSCrew

                                InstantaneousSCrew

                                ScrewANgle  ([/JCS],/PRO,/DIS)

                                TRaNslation     (/JCS,[/GCS],/PRO,/DIS)

                                MOmentArm       (/GCS,[/PRO],/DIS)

                                WRenchArm       (/GCS,[/PRO],/DIS)

EXPort      MaRKer      POSition     ([/GCS],/LCS)
                                DIStance
                                VelocityAndAccel

EXPort      SEGment     PaRaMeter   /ANImation

                                POSition    /MATrix
                                            /ANImation

                                VELocity        ([/GCS],/LCS)

                                ACCeleration    ([/GCS],/LCS)

                                ENeRgy

                                FiniteSCrew

                                InstantaneousSCrew

                                MOMentum

                                BodyCenterMass

                                BodyCenterMassVelocity

                                ANGle

                                SWIvel3D

EXPort      REConstructed   GroundReactionLoad { com, ft ([/GCS],/LCS) }

                                  CenterPRessure        { com, ft ([/GCS],/LCS) }

                                  StrikeIndex
```

Telio Command Summary

Commands for graph and stripchart display modes in Telio

Coordinate System names:

Acronym	Description
GCS	Lab global coordinate system
LCS	Segment local coordinate system
DIS	Distal segment local coordinate system (applies to joint quantities)
PRO	Proximal segment local coordinate system (applies to joint quantities)
JCS	Joint coordinate system
EUL	Fixed eular angle sequence

Summary of variable names:

Acronym	Description	Options
mpn	marker position	
mvl	marker velocity	
mac	marker acceleration	
spn	segment position	
smp	segment marker position	
smd	segment intermarker distances	
svl	segment velocity	(/GCS or /LCS)
sac	segment acceleration	(/GCS or /LCS)
ske	segment kinetic energy	
spe	segment potential energy	
ste	segment total energy	
sis	segment instantaneous kinematic screw	
sfs	segment finite kinematic screw	
smm	segment momentum	
san	segment angle to GCS	
bke	body kinetic energy	
bpe	body potential energy	
bte	body total energy	
bcm	body center of mass position	
bcv	body center of mass velocity	
bmm	body momentum	
jan	joint angle	(/EUL or /JCS)
jvl	joint velocity	(/EUL or /JCS)
jac	joint acceleration	(/EUL or /JCS)
jld	joint load	(/PRO, /DIS, /JCS, or /GCS)
jll	joint load- leg summed	(/JCS or /GCS)
jwr	joint wrench	
jpw	joint power transfer	
jpl	joint power transfer- leg summed	
jis	joint instantaneous kinematic screw	
jfs	joint finite kinematic screw	

jtr	joint translation	(/PRO, /DIS, /JCS, or /GCS)
jsa	joint finite screw angle	(/PRO, /DIS, or /JCS)
jma	joint moment arm	(/PRO, /DIS, or /GCS)
jwa	joint wrench arm	(/PRO, /DIS, or /GCS)
fpl	ground reaction forces on force plates	
fft	ground reaction forces on feet	
fcb	ground reaction forces on combined feet	
fcp	center of pressure of GRL of force plates	
fcf	center of pressure of GRL for feet	(/LCS or /GCS)
fcc	center of pressure of GRL for combined feet	
fwp	force wrench for force plates	
fwf	force wrench for feet	
fwc	force wrench for combined feet	
fp1	first integral of GRL upon force plates	
ff1	first integral of GRL upon feet	
fc1	first integral of GRL upon combined feet	
fp2	second integral of GRL upon force plates	
ff2	second integral of GRL upon feet	
fc2	second integral of GRL upon combined feet	
fsi	strike index for feet	
pma	pitching moment arm	(/PLV, /TRK, or /GCS)
pwa	pitching wrench arm	(/PLV, /TRK, or /GCS)
fcv	velocity of center of pressure in GCS	
rft	reconstructed ground reaction forces on feet	(/GCS or /LCS)
rcb	reconstructed ground reaction forces on combined feet	
rcf	reconstructed center of pressure of GRL for feet	(/GCS or /LCS)
rcc	reconstructed center of pressure of GRL for combined feet	
rsi	reconstructed strike index for feet	
emg	emg channel signal	
mln*	muscle length	
mvl*	muscle velocity	
mor*	muscle origin position	(/LCS or /GCS)
min*	muscle insertion position	(/LCS or /GCS)
mla*	muscle line of action	(/LCS or /GCS)
mem*	muscle/IEMG measure	

* not implemented yet...

tim	time (seconds)	
fra	frame number	
per	percent of gait cycle or data set	

Command parameter summary:

[] indicates parameters for graphing only
{} indicates parameters for 3d view only

Joint names:	rak	right ankle	lak	left ankle
	rkn	right knee	lkn	left knee
	rhp	right hip	lhp	left hip
	pvl	pelvis-lab		
	pvt	pelvis-trunk		
	nec	neck		
	rsh	right shoulder	lsh	left shoulder
	rel	right elbow	lel	left elbow
	rwr	right wrist	lwr	left wrist
Segment names:	rft	right foot	lft	left foot
	rcf	right calf	lcf	left calf
	rth	right thigh	lcf	left calf
	plv	pelvis		
	trk	trunk		
	hed	head		
	rua	right upper arm	lua	left upper arm
	rla	right lower arm	lla	left lower arm
	rhd	right hand	lhd	left hand

Acronym	1st param	2nd param	# graph param	# 3dvu param	
mpn	mrk#	[axis(x,y,z)]	2	1	
mvl	mrk#	[axis(x,y,z)]	2	1	
mac	mrk#	[axis(x,y,z)]	2	1	
spn	SegName	[axis(x,y,z)]	2	1	
[smp	SegName	mkr# axis(x,y,z)	3	-	]
[smd	SegName	dist#	2	-	]
svl	SegName	[axis(x,y,z,rx,ry,rz)] {type(rot,trn)}	2	2	
sac	SegName	[axis(x,y,z,rx,ry,rz)] {type(rot,trn)}	2	2	
[ske	SegName	comp(rot,tran,tot)	1	-	]
[spe	SegName		1	-	]
[ste	SegName		1	-	]
sis	SegName	[type(x,y,z,px,py,pz,rot,trn)]	2	1	
sfs	SegName	[type(x,y,z,px,py,pz,rot,trn)]	2	1	
smm	SegName	[axis(x,y,z,rx,ry,rz,rm,tm)] {type(rot,trn)}	2	2	
[san	SegName	axis(flex,abad,inex)	2	-	]
[bke	comp(rot,tran,tot)		1	-	]
[bpe			0	-	]
[bte			0	-	]
bcm	[axis(x,y,z)]		1	0	
bcv	[axis(x,y,z)]		1	0	
bmm	[axis(x,y,z,rx,ry,rz,rmag,tmag)] {type(rot,trn)}		1	1	

[jan	JntName	axis(fle,abd,int)	2			-	]
[jan/eul	JntName	axis(x,y,z)	2			-	]
[jvl	JntName	axis(fle,abd,int)	2			-	]
[jvl/eul	JntName	axis(x,y,z)	2			-	]
[jac	JntName	axis(fle,abd,int)	2			-	]
[jac/eul	JntName	axis(x,y,z)	2			-	]
jld/gcs,pro,dis	JntName	[axis(fx,fy,fz,mx,my,mz)] {type(rot,trn)}	2			2	
jld/jcs	JntName	[axis(mlf,apf,cdf,fem,aam,iem)]					
[jll/gcs	side(rt,lf)	axis(fx,fy,fz,mx,my,mz)	2			-	]
[jll/jcs	side(rt,lf)	axis(mlf,apf,cdf,fem,aam,iem)		2	-	]	
jwr	JntName	[type(x,y,z,px,py,pz,mom,frc)]			2	1	
[jpw	JntName		1			-	]
[jpl	side(rt,lf)		1			-	]
jis	JntName	[type(x,y,z,px,py,pz,rot,trn)]	2		1		
jfs	JntName	[type(x,y,z,px,py,pz,rot,trn)]	2		1		
[jtr/gcs,pro,dis	JntName	axis(x,y,x,mag)	2			-	]
[jtr/jcs	JntName	axis(mlt,apt,cdt,mag)	2			-	]
[jsa/pro,dis	JntName	axis(x,y,z)	2			-	]
[jsa/jcs	JntName	axis(fle,abd,int)	2			-	]
[jma	JntName	axis(x,y,z,mag)	2			-	]
[jwa	JntName	axis(x,y,z,mag)	2			-	]
fpl	Plate#	[axis(fx,fy,fz,mx,my,mz)] {type(mom,frc)}	2			2	
fft	side(rt,lf)	[axis(fx,fy,fz,mx,my,mz)] {type(mom,frc)}	2			2	
fcb	[axis(fx,fy,fz,mx,my,mz)] {type(mom,frc)}		1			1	
fcp	Plate#	[axis(x,y,z)]	2			1	
fcf	side(rt,lf)	[axis(x,y,z)]	2			1	
fcc	[axis(x,y,z)]		1			0	
fwp	Plate#	[type(x,y,z,px,py,pz,mom,frc)]	2		1		
fwf	side(rt,lf)	[type(x,y,z,px,py,pz,mom,frc)]	2		1		
fwc	[type(x,y,z,px,py,pz,mom,frc)]		1			0	
[fp1	Plate#	type(fx,fy,fz,mz)	2			-	]
[ff1	side(rt,lf)	type(fx,fy,fz,mz)	2			-	]
[fc1	type(fx,fy,fx,mz)]		1			-	]
[fsi	side(rt,lf)		1			-	]
[pma	axis(x,y,z,mag)		1			-	]
[pwa	axis(x,y,z,mag)		1			-	]
fcv	side(rt,lf)	[axis(x,y,z)]	2			1	

[rft	side(rt,lf)	axis(fx,fy,fz,mx,my,mz)	2	-	]
[rcb	axis(fx,fy,fz,mx,my,mz)		1	-	]
[rcf	side(rt,lf)	axis(x,y,z)	2	-	]
[rcc	axis(x,y,z)		1	-	]
[[rsi	side(rt,lf)		1	-	]
[emg	channel#		1	-	]
[mln	MuscleName		1	-	]
[mvl	MuscleName		1	-	]
mor	MuscleName	[axis(x,y,z)]	2	1	
min	MuscleName	[axis(x,y,z)]	2	1	
mla	MuscleName	[axis(x,y,z)]	2	1	
mem	MuscleName		1	1	

Overall Graph command form:

GRAph/#/SAM/DFX

:without DFX xvar{/OPT} {p1 p2} yvar{/OPT} {p1 p2} yvar {p1...

:with DFX xvar{/OPT} {p1 p2} yvar{/OPT} {p1 p2} xvar {p1 p2} yvar ...

Graph command options

/# number of graphs that will appear on screen/page

/SAM data sets in graph will use the same x axis

/DFX data sets will consist of independent x,y pairs

Graph variable options {/OPT}

/NGX Negate the x axis values of this variable's data set

/NGY Negate the y axis "

/NOA Don't align sides on this data set even if OPT GRA ALI ON has been set. This is a very special case which is used to handle the nomal joint angle data stored in NORM_ANG.ANZ which has been extracted from the original OSU OUT normal joint angle data set. These angles are stored with the right and left sides already time aligned in the gait cycle while every other data set that will be used here will have to have Telio align it's sides. This is a nasty little kludge but it works (see ANGLES.MAC, macro 4 for an example of this option being used).

/YTI Applies to EMG's only. Inserts the stored EMG channel name into the graph YTItle - especially useful for making EMG strip charts

/NON Don't normalize data for this variable even though normalization is set. This is used with normal joint moment and power data from the literature since it is in the ANZ files in normalized form already— it does not need to be normalized again...

/NON Don't normalize this quantity. This is mainly included so data which has already been normalized and stored in an ANZ file can be plotted on the same graph with data which requires normalization. Primary application is for normal moment, power and ground reaction load data sets taken from the literature and imported into ANALYZE. These are usually already normalized and must not be normalized a second.

IMPORTANT time saver:

GRA PRE will display the same set of data defined previously for a graph or set of graphs but reflecting most of the changes made in the graph options. This way you can taylor the graph(s) such as labels and the axis scales and then redisplay the graph without having to type the command line(s) required to display the graph(s). Also, works great for getting the display the way you want it and then changing the settings to print the graph and use the GRA PRE command. Then turn the print option off again.

Graph Display Options

General command form: OPT GRA/# cmnd switch/data
/# : indicates which graph number is being modified (default is 1)

Graph option commands: (values in [] are the default settings where applicable)

RES Reset all the options of the graph to their default values
CYC Clip data to gait cycle [OFF]
RAA Remove average angle (standing position joint angle) [OFF]
PRI Copy graph(s) being displayed into print file [OFF]
SCR Turn off/on the terminal screen (useful for fast printing a graph or when running Telio in batch mode to prevent the log file from becoming huge. [ON]
LGD Display legend in graph [ON]
MLG Display legend using manually input legend text [OFF]
EVE Display gait events [ON]
PCT Display percent cycle line rather marking gait events on the data points. This option only applies when the x axis of a graph is set to percent gait cycle. [ON]
ALI Align sides. Rearrange the data from the side of the second heelstrike such that when the data is graphed, it looks as if the two sets have the same time scale rather than being offset by a half a cycle. [ON]
LIN Display graph data sets with lines between the data points [ON]
SYM Display the graph data points with symbols [ON]
GRL Display the graph label either manual or automatic versions [ON]
MGL Display the manually input graph label. The automatic graph label is the text in the patient name field of the ANZ file. [OFF]
NWT Normalize forces, moments, powers to body weight [OFF]
NWH " " " " " " " *height [OFF]
NWL " " " " " " " *leglength [OFF]
XZL Display the x axis zero line [ON]
YZL Display the y axis zero line [ON]
PFZ Make the fz value of the GRL positive [ON]
SAS Use simple autoscaling for determining the extents of the graph axes. This will make the scales +/- 10% of the min/max of the x and y data. The normal scaling method tries to be fancy and find increments for the axes that display nicely, but it blows it sometimes. Thus, this option provides a way out if it goes wrong. [OFF]
FGE Display only the first data set's gait events. Normally the gait events for every data will be displayed, but for different gait runs with different timings of the events the graph can become quite messy. Thus, this allows the graph to be cleaner by excluding information. [OFF]
MSC ON/OFF Use manually defined scale limits for the graph [OFF]
XMIN XMAX YMIN YMAX Set the scale limits [0 100 0 1]
DEF Default file number from which data will be taken for the graph [1]

LGP Legend position 1-9 1: Bottom left 9: Top Right [1]
DIV Number of divisions for the x and y axes [5,5]
FON Number of font being used for text in graph [1]
MLT Input manual legend text
MGT Input manual graph name text
TIT Input graph title Use *** to make entry contain no text
SUB Input graph subtitle
XTI Input x axis title
YTI Input y axis title

NOTE: A special graph form may be invoked to print out the time-distance parameters in the space normally occupied by a graph. It can be used by stating GRA BLA in which case a blank graph area is produced. When this command has been given it may be followed by these keywords: Indiscriminate use of the following commands, especially TD1 and TD2 on the same graph, will make a mess.

TD1 First set of time distance parameters written to graph area (these are the distance measures)
TD2 Second set of time-distance parameters written to graph area (these are the time parameters)
TIT Title of graph will be inserted in graph area
SUB Subtitle inserted
NAM Name of patient placed in graph area
DAT Date of analysis placed in graph area
FIL Name of data file containing data for graph will be written to graph area

Overall StripCHart command form

sch/# xvar {p1 p2} yvar {p1 p2} yvar {p1 p2} ...

Each of the graphs in the strip chart is defined using the same command syntax that is used with the GRAph command.

IMPORTANT time saver:

SCH PRE will display the same set of data defined previously for a graph or set of graphs but reflecting most of the changes made in the graph/chart options. This way you can taylor the graph(s) such as labels and the axis scales and then redisplay the graph without having to type the command line(s) required to display the graph(s). Also, works great for getting the display the way you want it and then changing the settings to print the graph and use the GRA PRE command. Then turn the print option off again.

Stripchart option commands

NOTE: Each individual graph in the stripchart is actually controlled by the options that apply to the graphs. To customize the appearance of the individual sections of the graphs, the OPT GRA commands must be used. The OPT SCH commands affect the overall appearance of the stripchart.

EVE Display the gait event lines on the stripchart [ON]
TIT Set the title for the stripchart
LAB Show the stripchart label that appears in the lower left corner [ON]
MLB Use the manually set stripchart label rather than the automatic label which is set to the patient name in the ANZ file ON/[OFF] or Label text
MNX Use/set the x axis of the stripchart to a manually defined range of values ON/[OFF] or xmin xmax
SAS Set the autoscaling of the stripchart x axis to be the simple autoscaling described in the graph options
PRI Print the stripchart [OFF]
SCR Turn the terminal display [ON]/OFF
SUB Set the stripchart subtitle

XTI	Set the x axis title
XDV	Set the number of divisions on the x axis scale [5]
FON	Set the font used for the strip chart text [1]

STAtus commands

There are a bunch of status commands which can be used to check what a graph or stripchart's settings currently are and to look at what data will be used in the plots.

Show3D Command Summary

Command	Param1	Param2	Param3
REAd	ANZ filename		
3DSingle	SIDe TOP FROnt ISometric1 ISometric2 MANual	MaRKer SEGment AVeraGe	Segment Name/RT/LF/ALL
3DMultiple	SIDe TOP FROnt ISometric1 ISometric2 MANual view direction	MaRKer SEGment AVeraGe	Segment Name/RT/LF/ALL
OPTion	WIRe/NOWire frame of segment CENter/NOCenter display of segment ERAse/NOErase previous frames STEp/ANImation through data frames LCS/NOLcs display segment LCS FACe/NOFace remove hidden faces of segment HIDe/NOHide remove segment hidden lines CSY/NoCoordinateSystem show lab GCS PRInt/NOPrint send screen image to disk file ManualWiNdow/AutomaticManualWindow use manually defined view window ManualVieW define the manual view direction SeGmentCentered/NoSG **Segment Name** center view on segment LCS center STArt frame # INCrement # between frame increment		
STAtus			

ANZ Calculation Dependencies

Command		Required Previous Commands
AVG	POS	SEG POS
	ANG	JNT JAN and/or JNT EAN
	REF	AVG POS
CYC	EVE	REA ANZ or TR3 or TRU file
	STA	SEG POS CYC EVE
EMG	NAM	REA GLS1 or GLS2 or FPD file with emg data
	FIL	REA GLS1 or GLS2 or FPD file with emg data
	REC	REA GLS1 or GLS2 or FPD file with emg data
	INT	REA GLS1 or GLS2 or FPD file with emg data
	MWA	REA GLS1 or GLS2 or FPD file with emg data
FRC	CON	REA GLS1 or GLS2 or FPD file with force plate data
	ALI	FRC CON CYC EVE
	CLI	ALI
	CPR	FRC CON FRC ALI if center of pressure under individual feet is desired
	LCP	FRC ALI FRC CPR SEG POS SEG PRM
	WRE	FRC CPR FRC ALI if GRL wrench under individual feet is desired
	INT	FRC CON FRC ALI if GRL integral under individual feet is desired
	REC	JNT LOA

FRC	PMA	FRC CLI SEG BCM FRC CPR
	PWA	FRC CLI SEG BCM FRC WRE
	CPV	FRC LCP SEG VAA
	FIL	REA GLS1 or GLS2 or FPD file with force plate data
JNT	JAN	SEG POS
	JVA	JNT JAN SEG ANG SEG VAA
	EAN	SEG POS
	EVA	JNT EAN SEG VAA
	LOA	SEG POS SEG VAA SEG PRM JNT JAN FRC LCP if GRL data is to be used in calculation
	WRE	JNT LOA
	POW	JNT LOA JNT VAA
	TRN	SEG PRM SEG POS JNT JAN
	SAN	SEG FSC JNT JAN
	MAR	JNT LOA
	WAR	JNT WRE

MRK	INT	REA marker data from ANZ or TR3 or TRU file
	ADJ	MRK INT
	SMO	MRK ADJ
	FIL	REA marker data from ANZ or TR3 or TRU file
	REG	MRK ADJ CYC EVE if gait cycle data is to be used for regularizing
SEG	PRM	MRK ADJ
	MRK	SEG POS
	POS	MRK ADJ AVG REF if SEG POS/REF is given
	VAA	SEG POS MRK SMO/VAA if SEG VAA/MRK is given SEG MRK if SEG VAA/MRK is given
	ENR	SEG PRM SEG VAA
	FSC	SEG POS
	ISC	SEG VAA
	BCM	SEG PRM SEG POS
	MOM	SEG PRM SEG VAA
	ANG	SEG POS

ANZ Site Specific Data

Site/System Specific data required by ANZ

Motion system Coordinate system (lab GCS)

Minor adjustment may be need in the marker coordinates as they are read in from the file. More than probably just a rotation of CS's and/or add/subtract a distance to/from one or more component. ANZ's internal CS assumes a right hand system where x is in the front/back direction y medial/lateral and z vertically upward. The center of the CS is on the surface of the floor between the force plates. NOTE: There is a correction applied to the TR3 file marker data in the present ANZ code because OSU's VICON was setup years ago with the x direction medial/lateral and the y direction front/back plus the center of the CS is 19 inches above the floor !!! (I didn't do it- I only work there :-) Look at the code in read_tr3_file which is contained in file.for...

Force plate plate:

Placement/orientation relative to lab GCS

Change data in anz_init_constants.inc for the variable PltLoc. The plate position/orientation relative to the lab GCS is stored here as a 4x4 matrix for each plate. The initial 3x3 is the rotation matrix of the plate internal CS relative to the lab GCS. The last column is the position of the plate CS relative to the lab GCS.

Amplifier gains and/or calibration matrix

Change data in anz_init_constants.inc for the variable CalMatrix This is a 6x6 calibration matrix for each force plate. The channel gains as well as their excitation voltages are incorporated into the matrix. Thus, the raw force plate voltages are all that are needed as input to these matrices. These matrices are used in the routine that reads GLS2 files (the OSU electrical data file format). There is provision to insert normalized calibration matrices into CalMatrix and the amp gains into NewAmpGains. Basically this section will probably need a rewrite for the specific site. NOTE: FPD files already incorporate calibrated force plate data (I think so at least) and these are already supported by ANZ. This provides a way around having to rewrite the force plate reading section of ANZ.

Physical size

The width and length of the plates must be changed in anz_init_ constants.inc. The relevent variable is PltSize. These are only used in TELIO to allow the representation of the force plates in the 3D views.

Marker system

Right now only 2 marker sets are supported in ANZ: 1) The OSU full body marker set (21 markers) and 2) An upper body only marker set incorporating only the right arm which was part of a preliminary research project at OSU. Changing the marker system used is not a task for the weak of heart but is definitely possible. A separate write up on how to do this will be completed in the future. By the way TELIO is quite marker system independent and should not require any changes at all to work with another marker system. (No changes were made when the upper extremity system was added!)

ANALYZE source file/Routine name Cross Reference

Source File Name	Routine Name
Anz.for	Anz SetOptions FlipLogical GetPaitientInfo
Anz_File.for	Read_anz_file WriteStatus General_itor Force_itor Segment_itor Cycle_itor SwapInteger SwapIntegerArray SwapLogical SwapLogicalArray Read_1dim_array Read_2dim_array Read_3dim_array Read_4dim_array Zero_motion_flags Zero_force_flags Write_anz_file CountNumVarSaved IncVarCnt General_rtoi Force_rtoi Segment_rtoi Cycle_rtoi Write_1dim_array Write_2dim_array Write_3dim_array Write_4dim_array Find_scale_factor_1d Find_scale_factor_2d Find_scale_factor_3d Find_scale_factor_4d
Anz_init_anl.for	InitFullBodyAnalysis InitUpperExtremAnalysis
Anz_jnt_ang_ref.for	CalcJCSAnglesRef FindJntRefAng JCSAnglesRefAng RotMattoJcsAng JCSAngtoRotMat ReadSetRefAng CalcAnatSegPosRef
Average.for	Do_AVG_calc
Char.for	ParseLine MakeUpperCase MakeReal (f)

	RemoveCMNDOpt
Cycle.for	Do_CYC_calc InputGaitEvents CheckGaitPattern (f) CalcGaitStatistics
Digital.for	DigFilterDataSet FilterEMG FilterFRC BuildFilter Realft Four1 Swapr
Emg.for	Do_EMG_calc RectifyEMG IntegrateEMG MoveWinAvgEMG
Export.for	ExportData ExportAVG ExportCYC ExportREC ExportEMG GetSegNum (f) GetJntNum (f) GetMrkNum (f) GetCSType (f) OpenTextFile (f)
ExportFrc.for	ExportFrc ConvGRL ExpFrcGRL PrintGRLHeader ExpFrcCPR ExpFrcLCP ExpFrcWre ExpFrcSI ExpFrcInt1 ExpFrcInt2 ExpFrcPma ExpFrcPwa ExpFrcCpv
ExportJnt.for	ExportJnt ConvertNormJntLoad ConvertNormJntPower ExpJntJCSAng ExpJntJCSVAA ExpJntEulAng ExpJntEulVAA ExpJntLoad PrintJntLoadUnits PrintIgorJntLoad ExpJntLegLoad PrintIgorLegLoad ExpJntWre

	ExpJntPower	
	PrintJntPowerUnits	
	ExpJntLegPower	
	ExpJntFinScrew	
	ExpJntInsScrew	
	ExpJntScrewAng	
	ExpJntTrans	
	ExpJntMomArm	
	ExpJntWreArm	
	ExpJntAngRefCorAng	
ExportMrk.for	ExportMrk	
ExportSeg.for	ExportSeg	
	BuildFileNames	
	ExportSwivel3D	
	ExpSegPos	
	ExpSegVel	
	ExpSegAcc	
	ExpSegAng	
	ExpSegEnergy	
	ExpSegFinScrew	
	ExpSegInsScrew	
	ExpSegMom	
	ExpBodyCM	
File.for	Read_file	
	Get_file_name (f)	
	Read_c3d_file	
	AssignFullAnlMrkName	
	Read_tr3_file	
	Read_tru_file	
	Read_fpd_file	
	Read_gls_file	
	Read_gls2_file	
	ReadGls2Bytes	
	Save_file	
	Save_gls2_file	
	CalcScaleFactor	
	WriteGls2Bytes	
	FillGls2Header	
	Save_tru_file	
Filter.for	Filter	
	FiniteDif	
	DPF	*
	SWITCH	*
	MACRO	*
	SIGMA	*
	TDYN	*
	INVSP	*
	SPOFA	*
	SDOT (f)	*
	SPOSL	*
	SAXPY	*
	TRACE	*
	GCVSPL	**
	BASIS	**

	PREP	**
	SPLC (f)	**
	BANDET	**
	BANSOL	**
	SEARCH	**

* Authors: Hank Busby & David Trujillo
** Author: Herman Woltring

Force.for	Do_FRC_calc
	ConditionFrcData
	AlignFrcMtnData
	SetTrigFrm
	SetFtPlt
	ClipFrcMtnData
	Swap
	CalcCntrPres
	CalcCntrPresLCS
	CalcFrcWrench
	CalcFrcIntegral
	IntegrateData
	ReconstructGRL
	CalcPitchMomentArm
	CalcPitchWrenchArm
	CalcCprVel
Joint.for	Do_JNT_calc
	CalcJCSAngles
	CalcJCSAnglesSimple
	HandleAngleError
	InterpAngles
	CalcJCSVelAcc
	CalcEulVelAccYXZ
	AngFilterStatus
	CalcEULAngles
	CalcEULVelAcc
	CalcJntLoad
	CalcAllJntLoads
	CalcAJntLoad
	LoadLCStoLCS
	LoadGCStoLCS
	LoadLCStoGCS
	LoadGCStoJCS
	CalcJntWrench
	CalcJntPower
	DigFilterJnt
	CalcJntTranslation
	CalcJntScrewAngle
	CalcJntMomentArm
	CalcJntWrenchArm
Marker.for	Do_MRK_calc
	InterpolateMrkData
	Bad_marker (f)
	Find_interp_frm
	AdjustMrkData
	Lagrange_Interp
	SmoothMrkData

	DigFilterMrkData RegularizeMrkData
Math_sub.for	Gauss_Elim MatInv (f) VecChangeCS PntChangeCS DistPtPlane DistLineLine DistPtLine VecDot VecMag VecSub VecX VecNorm VecLth PntGCStoLCS VecGCStoLCS VecLCS1toLCS2 MatTran MatMult MinLineXPnts EulerAngles EulerAngles2 ShldrAngles JCSAngles SwivelAngles TriCen Line_Xsect PerpenPtLine
Segment.for	Do_SEG_Calc CalcSegParams CalcFullBodySegShape CalcFullBodyMassInertia CalcFullBodySegEnds CalcFullBodySegPos CalcLCSFoot CalcLCS1 CalcLCSCalf CalcLCSPelvis CalcLCSTrunk CalcLCSHead HandleSegPosError InterpSegPos CalcSegFiniteScrews CalcFSAWaldron CalcFSAWoltring CalcSegInstantScrews CalcInstantScrewAxis CalcSegEnergy CalcMrkDiagnostics CalcBodyCM CalcSegMomentum CalcSegGCSAngles DigFilterSeg
SegVelAcc.for	CalcSegVelAcc

	RotMatDeriv MakeOrthoNorm OldEulAngDeriv YXZEulAngDeriv CalcSegRotVelorAcc SegFilterStatus1 SegFilterStatus2 MrkVAAtoSegVAA FormSolveLSEqtn
Status.for	DisplayStatus DisplaySegment DisplayCycle DisplayAverage
Upper.for	CalcUpperBodySegPos CalcLCSPelvis2 CalcLCSTrunk2 CalcForeArmLCS CalcLCSHead2 CalcUpperBodySegEnds CalcUpperBodySegShape

Updating ANZ Source Code

Steps required to add a variable to ANZ

NOTE: Before modifying any of ANZ's source code **make a back up !!!!**

1) Add variable to desired common block

   Ex: BCMVel added to common /Segment/ in anz_seg.inc

2) Add calculation flag structure

   Ex: SEG_FLG.BCMVel added to structure /segment_flag/ in anz_structures.inc

3) Add indication of calculation status to status routine in status.for

4) Update structure /All_Flags/ in anz_file.inc Primarily this is the comment about the # of bytes in the flag structure to which the variable has been added.

5) Add pointer location for new variable to structure /anz_layout/ which is contained in anz_file.inc Make sure that it is added at the end of the list of pointernames

   Ex: SEG_BCMVel is added after FLAGS

6) Update MaxVar in write_anz_file stored in anz_file.for MaxVar is the maximum number of quantities that are possible to be stored in the ANZ file.

   NOTE: Right now there is only room in the structure /anz_layout/ for 128 separate variables and ~108 are being used now. The restriction comes from using 512 byte records for reading and writing which can, as a result, only contain 128 integer*4 numbers and a single record at present is being used to store the pointer structure. If more than 128 variables are to be stored than some additional code must be added to ANZ to handle more than a single record holding the variable location/scaling pointers...

   Ex: Increase MaxVar from 107 to 108

7) Update the routine write_anz_file by adding the code to save the new variable

   Ex: Added code to write BCMVel immediately following code to write BodyCM. The code was essentially a copy of that to write out BodyCM just modified to save the new variable.

8) Noting position of new variable writing code add a corresponding increment to CountNumVarSaved in anz_file.for This allows a display count of the progress of the file saving process.

   Ex: Added the following line right after the line of BodyCM

   ```
   if ( SEG_FLG.BCMVel ) Call IncVarCnt( 108,VarNum(NumVar+1),NumVar )
   ```

   NOTE: That 108 is the location of the new variable pointer in structure /anz_layout/

9) Increase the size of AnzVarName to size of MaxVar. AnzVarName is contained in common /Names/ which is in anz_names.inc

Ex: Increase from 107 to 108

10) Add AnzVarName initialization text to anz_init_constants.inc at the end of the last data statement in the file. This is used to display the name of the variable being written/read.

Ex: Add 'Body CM Velocity ' after 'Calculation Flags '

NOTE that the variable name must be 20 characters long.

11) Add code to read_anz_file to read new variable from ANZ file

Ex: Add code to read BCMVel immediately following the code to read BodyCM

12) Update code at front of read_anz_file which is used to convert between least significant byte to most significant byte computer storage methods for the calculation FLAGS. This is rather messy since it is dependent upon how the calculation flags were ordered in the structure /All_Flags/ and the fact that only the logical*4 and integer*4 quantities in the structures are to have their bytes reversed if the ANZ file is moved between computers.

Ex: Since BCMVel was added to SEG_FLG which is in the middle of the FLAGS structure the position of last logical flag of EMG_FLG was increased by 1. The numbers within the code at 'if (ptr.FLAGS .ne. 0) then' were changed. Thus,

```
call SwapLogicalArray( 55,FLAGS.array )
do i= 66, 70 ! EMG flags
i= FLAGS.array(71) ! EMG.FLG.NumProcOp
call SwapInteger( i,FLAGS.array(71) )
```

changed to

```
call SwapLogicalArray( 56,FLAGS.array )
do i= 67, 71 ! EMG flags
i= FLAGS.array(72) ! EMG.FLG.NumProcOp
call SwapInteger( i,FLAGS.array(72) )
```

This is very messy and must be done with utmost caution. Moving things around too much will make previous ANZ file's calculation FLAGS be read in incorrectly. The data will still be read from ANZ file but some of the nice save features such as the operations on the EMG data and indications as to smoothing/filtering of various data will be hopelessly scrambled.

12) Update the zero_motion_flags or zero_force_flags, which ever is appropriate, in file.for.

Ex: Add SEG_FLG.BCMVel= .false. to zero_motion_flags

13) Add export variable code to appropriate export routine...

Ex: Add ExportBCMVel to ExportSeg.for

Updating Telio Source Code

Updating TELIO to display new variables (graphing functions only):

1) Add pointer location for new variable to structure /anz_layout/ which is contained in tel_anz_file.inc Make sure that it is added at the end of the list of pointer names

Ex: BCMVel is added after FLAGS

2) Add the variable name text to and increase the size of the array AnzVarName which is defined in tel_names.inc and is initialized in tel_init.inc.

Ex: Add the text 'Body Cntr Mass Vel ' to tel_init.inc
Increase AnzVarName(107) in tel_names.inc to 108

3) Add variable name/number to tel_const.inc If you do not add this at the end of the list of numbers than all the numbers must be shifted after the one you insert.

Ex: Add parameter BCV_DATA= 21
Shift all variable numbers up by 1 starting BMM_DATA on

If numbers are shifted, then the routines which identify a command to a specific class of data must be updated. The classes are:

Class	Subroutine
Marker	MrkCmnd
Segment	SegCmnd
Joint	JntCmnd
Force	FrcCmnd
Reconstructed	RecCmnd
EMG	
Average	
Muscle	MslCmnd
Time	TimeCmnd

These routines are in the file buildgraph.for

4) Add the 3 letter acronym for the new variable to the listing of acronyms in the array VarNames. The order of the variable names MUST be the same as the variable name/numbers in tel_const.inc. The acronyms for VarNames are initialized by a data statement in the file tel_init.inc Also, the dimension of the VarNames array must increased in tel_names.inc

Ex: Add 'BCV' immediately following 'BCM' in tel_init.inc
Increase the dimension of VarNames from 69 to 70

5) Modify the flags structure of the variable class so that there is an indicator of whether it has been read. These structures are defined in the file tel_struct.inc

Ex: Add .BCMVel to the SegmentFlag structure

NOTE: The ANZ file contains a copy of the flags and thus the structures here must be identical to those used in ANZ if these flags are to be read properly.

6) Modify the code in the subroutine ZeroFlags, contained in the file tel_file.for, so that the new flag position is zeroed properly.

Ex: Increased the do loop zeroing the FD.SF.Flag array from 11 to 12

7) Modifiy the code in ReadANZFileParam, contained in the file tel_anz_file.for, so that the flags are read properly and if there are any initialization variables that must be read. If only a change in the flags as occured, then no change to this code should be necessary because the changes in the flag structures in step 4 should takecare of things. The reading of the flag structure from the ANZ file accomplishes this task...

Ex: No modification of ReadANZFileParam was needed to add BCMVel

8) There is an error trap (format statement # 100) in AssignPlotData that displays all the variable names if a mistake is made in the command line when a variable name should be given. This write statement has the total number of variable names incorporated into it. Thus, when the dimension of the array VarNames is increased, the number of variables printed out by this statement must be increased to this same value. AssignPlotData is in buildgraph.for

Ex: Increase k=1, 69 to k=1, 70 for this statement

9) Update the arrays GrNumPrm and TdNumPrm defined in tel_graph.inc and tel_3dvu.inc respectively and initialized in tel_init.inc These arrays contain the number of command line parameters that must appear after the variable name for the data to be correctly defined for graphing or 3d display. The dimension of the arrays must be increased to be the same as the dimension of VarNames. The order of data elements in the initialization data statement must be the same as the VarNames 3 letter acronyms.

Ex: Add ,1 to the GrNumPrm initialization immediately following the number of paramters for the BCM variable and add ,0 to TdNumPrm in the same place. Increase the dimension of these arrays from 69 to 70. NOTE: The GrNumPrm value is 1 because the component (x,y,z) of the Body CM Velocity must be given immediately following the BCV variable name on the command line. In the case of TdNumPrm, no furhter information is needed since the velocity is drawn as a vector originating from the BCM point.

10) Add code in the respective Assign data routine for the class of variable added to make it possible to read data in from ANZ file. The routines corresponding to the data classes are:

Class	Subroutine
Marker	AssignMrkData
Segment	AssignSegData
Joint	AssignJntData
Force	AssignFrcData
Reconstructed	AssignRecData
EMG	AssignEmgData
Average	
Muscle	AssignMslData
Time	AssignTimeData

These routines are in the file buildgraph.for The number and type of parameters passed into the routine ReadFileData are critical to assuring proper reading of the data, so be careful here. Otherwise the program will more than probably read something in from the file but as soon it gets graphed you'll see that it is gibberish. (ReadFileData routine is in tel_file.for) Be careful about the number of command line parameters needed by variable. These are defined in GrNumPrm

Ex: Add code to AssignSegData to read BCV immediately following the code for reading BCM. Use BCM code as a prototype for BCV. In fact in this case the two sections are identical.

11) Add the code which reads the data from the ANZ file. This is added to the routine ReadANZFileData which is in tel_anz_file.for This code is very dependent upon the exact shape of the array in which the data is stored in the ANZ file. Look at the explaination with the routine for more details. Typically it is best to use a variable whose array shape is similar to the new variable as a prototype for the new code.

Ex: Add code to ReadANZFileData to read BCV immediately following the code for reading BCM. Use BCM code as a prototype for BCV.

12) Add variable to status display so that user can tell whether the new variable is present in the ANZ file.