(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 395 773 A1**

## (12) EUROPEAN PATENT APPLICATION

(21) Application number: 89107924.6

(51) Int. Cl.$^5$: **G06F 15/28**

(22) Date of filing: 02.05.89

(43) Date of publication of application:
07.11.90 Bulletin 90/45

(84) Designated Contracting States:
AT BE CH DE ES FR GB GR IT LI LU NL SE

(71) Applicant: ESRAC COMPUTER CORPORATION
131 Spring Street 6th Floor East
New York New York 10012(US)

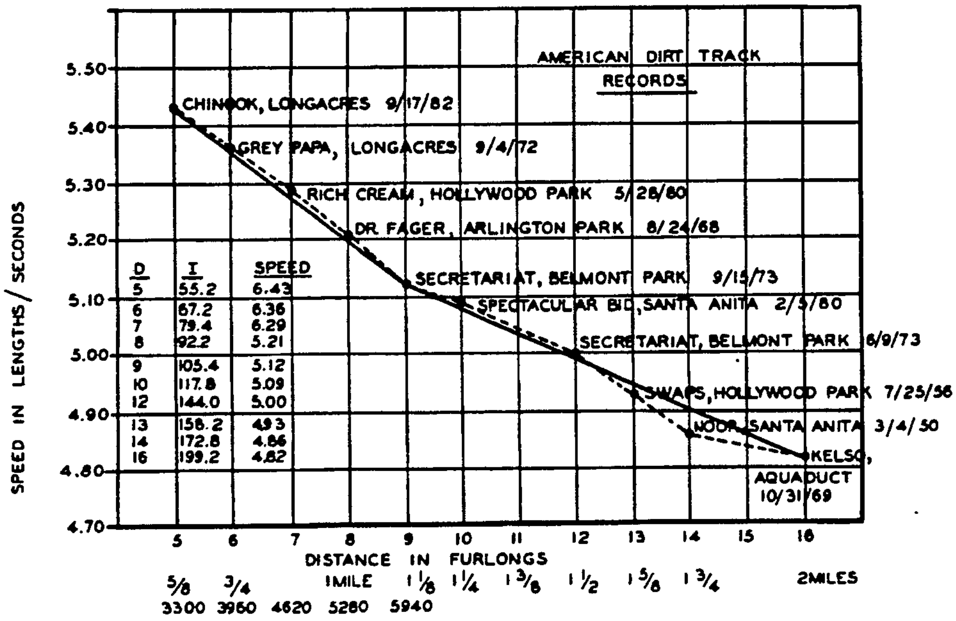
(72) Inventor: Sinn, Robert S.
248 West 88th Street Penthouse B
New York New York 10024(US)

(74) Representative: Wasmeier, Alfons, Dipl.-Ing. et al
Postfach 382 Greflingerstrasse 7
D-8400 Regensburg(DE)

# (54) Electronic speed rating calculator and method.

(57) Apparatus and method are set forth for calculating a comparative speed rating for an entrant in a race such as a horse race so that the entrants' performance can be compared with the performance of other entrants. The speed rating is determined in accordance with a formula which is based on a relationship between speed and distance over a particular distance for a particular class of entrant taking into consideration the track and its conditions; the class of track, the race type allowances and the age of the horse and time of year.

FIG. 1



| D | T | SPEED |
|---|---|---|
| 5 | 55.2 | 6.43 |
| 6 | 67.2 | 6.36 |
| 7 | 79.4 | 6.29 |
| 8 | 92.2 | 5.21 |
| 9 | 105.4 | 5.12 |
| 10 | 117.8 | 5.09 |
| 12 | 144.0 | 5.00 |
| 13 | 158.2 | 4.93 |
| 14 | 172.8 | 4.86 |
| 16 | 199.2 | 4.82 |

EP 0 395 773 A1

# ELECTRONIC SPEED RATING CALCULATOR AND METHOD

## BACKGROUND OF THE INVENTION

The present invention relates to methods and systems for calculating speed rating for paticipants in a speed contest or race. More particularly, the present invention relates to methods, systems and formula for determining a speed value or speed rating for participants, such as race horses, in a speed contest, such as a horse race, for example.

There are many methods for calculating a speed rating for race horses. A speed rating of a race horse is a comparative rating of the speed of the particular race, taking into account the distance, in furlongs, of the race.

An animal's ability to race is a function of this innate ability modified by its physical condition at the time of the race. That is, an animal in peak physical condition can run over a given distance at a certain average speed, taking into account that the particular animal is, at the time, running the distance at its ultimate capacity and no further amount of training can improve its performance. The innate ability to race for the individuals of any species has a bell shaped curve, the same as all other physical and performance characterstics, such as height or intelligence. The individuals of a species tend toward a normal, that is, the peak of the bell shaped curve, with exceptional individuals of the species out at the tails on either end of the bell.

For many years, horse racing enthusiasts have been seeking a method to evaluate the innate running capacity of horses, at various distances and at different race tracks, different conditions, different races, and takes horses of different sex and age. This is difficult for several reasons:

a) Horses race at many different distances at different tracks and there is great difficulty in relating the performance in one race at one track at one distance to another performance in another race at another track at another distance.

b) Although past performance of all horses in a race are published in a publication called the Daily Racing Form, this does not take into the class of track of other factors.

c) The same horse will run at different speeds on different tracks. This difference is a function of the track structure, ie., the length of the track and the track condition. There are both long term variations. Long term variations are a function of the track structure, while short term variations are a function of weather, ground conditions, amount of scraping, etc.

The racing class of a horse is related to his gposition on the bell shaped performance curve. A horse of high class, a horse whose position on the curve is out on the high side of the bell, will beat a horse of average class, a horse whose position on the curve is at the peak of the bell at any typical racing distance. Higher class horses appear to have more ability and tend to perform better than lower class horses at all racing distance. Higher class horses appear to have more ability and tend to perform better than lower class horses at all racing distances.

The problem is how to rate a horse's ability such that:

a) Horses of the same class average the same rating at all distances.

b) Horses of different class have different ratings on an ascendant scale with performance.

c) Horses taken individually on the average have the same speed rating at all distances.

Some of the above problems have found solution in the teachings of U.S. Patent No. 4,133,031 and a pending United States Application Serial No. 789,123 whereas the speed of a race horse is calculated by a method that employs a relationship between speed and distance for horses of the same class, a relationship between horses of different classes and a relationship between speed and distance for a particular horse, and where taking into consideration those relationships a specific equation is employed to determine the winner. The equations taught are as follows:

For 9 furlongs or less:

$$SR = \frac{\frac{(f \times 60)}{(t)} + 0{,}077f - 4{,}92}{0{,}008}$$

For more than 9 furlongs:

$$SR = \frac{\frac{(f \times 60)}{(t)} + 0{,}0465 \times f - 4{,}686}{0{,}008}$$

4

Where f is the length of the race in furlongs.
Where t is the time of the winning horse, in seconds.

It has been found that this equation is substantially accurate and effective for deriving the speed rating (SR). However, this equation lacks certain factors, type of race, price, speed, sex, age and class of track. The present system seeks to take these into consideration by adding track variants according to the formula:

$$TV\ (T,D,C) = \frac{\sum_{i}^{N} [CRRi(T,D,C)-AL(CD(T,D,C))]}{N}$$

Where TV is the track variant.
Where T is time.
Where D is date.
Where C is category.
Where N is number of races in that category.
Where i is the running index of races in that category.
Where CRR is connected race rating.
Where AL is algorithm.
Where CD is connections used on type of race.
Where CRR = SR (Winner) + Sex Corrective + SBC
Where
CD = OCL 1 x Price
BCL .8 x Price
MCL .5 x Price
SAL 2 x Price
SHC 2 x Price

## SUMMARY OF THE INVENTION

The present invention provides takes into consideration the equations for solving the speed rating (SR) of a horse. The equations are:

$$SR = \frac{\frac{(f \times 60)}{(t)} + 0{,}0465 \times f - 4{,}686}{0{,}008}$$

and

$$SR = \frac{\frac{(f \times 60)}{(t)} + 0{,}0465 \times f - 4{,}686}{0{,}008}$$

and adds to them factors for type of race, space, applicable, purse, sex, age and class of track according to the formula

$$V(T,D,C) = \frac{\sum_{i}^{N} [CRRi(T,D,C)-AL(CD(T,D,C)]}{N}$$

Accordingly, the method of the present invention includes the steps of classifying the entrant relative to other entrants in the same or similar sporting events. The speed rating of the entrant is calculated according to a mamthematical formula which includes the pertinent factors for determining a speed rating for each entrant independent of variations such as track and classification of the entrant. Thus, by the method of the present invention, a speed rating is calculated which is general in its application in that the same speed rating applies for the entrant regardless of the distance of the track on which the race is run.

The apparatus of the present invention includes means for data entry such as keyboard similar to keyboards commonly used by 4-function arithmetic calculators, a random access memory for storing input data and other intermediate and output data, and arithmetic unit for calcuting a speed rating given the input data, a display device for displaying selected data and a read only storage program control for controlling the sequence of steps to be executed in the calculation of a speed rating.

These and other objects, features, and advantages of the present invention, together with the operation of the invention, will be understood by reference to the following detailed description taken together with the following drawings.

## DETAILED DESCRIPTION OF THE DRAWINGS

Fig. 1 illustrates the generator of the speed rating system.

Fig. 2 is a graph illustrating the relationship between Race Rating and Corrected Dollars for an idealyzed medium fast track.

Fig. 3 is a chart showing an example of the invention.

Fig. 4 is a graph of Race Rating versus Corrected Dollars.

## PREFERRED EMBODIMENT OF THE INVENTION

For many years horse racing enthusiasts have been seeking a method to evaluate a horse's innate running capacity at various distances. This is difficult for several reasons:

1. Horses race at many different distances and there is great difficulty in relating performances at the different distances. For example, if you know the racing ability of horse B at 1 1/2 miles and the racing ability of horse A at 1 1/2 miles and the racing ability of horse A at 1 mile, who would be the fastest at 1 1/4 mile?

2. The same horse will run at different speed on different tracks. This difference is caused by the track structure and the track condition. There are long term variations (track structure) and short term variation (weather) and etc.

A horse's racing class is related to this position on the bell shaped performance curve. A horse of high class (out on the high side tail of the bell) at any typical racing distance. Higher class horses tend to perform better than lower class horses at all racing distances.

The problem is how to rate a horse's racing ability such that:

1. Horses of the same class average the same rating at all distances.
2. Horses of different classes have different ratings on an ascendant scale with performance.
3. Horses taken individually on the average have the same speed rating at all distances.

The present speed rating systems do not meet the criteria as stated above. In most systems such as the Daily Racing Form one point is subtracted from 100 for each fifth of a second the horse's performance was higher than the track record at that distance. This system does not meet the criteria stated above for the following reasons:

1. One fifth of a second at a distance is much less than one fifth of a second at a sprint.
2. The track records at different distances could have been set by horses of different classes. The track record is a function of the horse that set the record.
3. The rule of thumb of one fifth second equals one length is not accurate.

In all the present systems of speed ratings, the focus is on the time of the race. The basic discovery of the linear relationship of speed and distance for horses of the same class now allows for a system which fulfills the original requirements of the speed rating system. The slope of the line relating North American Records was found to be about exactly the same as the slope of the line relating horses of all different classes of horses. This slope is approximately (0,07) length/second furlong below 9 furlongs and approximately (0,0465) length/second above nine furlongs.

Fig. 1 illustrates the generator of the speed rating system. The ordinate is the average speed over the distance. The abcissa is the distance in furlongs. The desire is to make a scale of 0-100 where the vast majority of all performances will fall within that scale. The rating of 100 is chosen as follows:

An optimum speed at nine furlongs is five lengths per second (this is approximately the North American Record. This is assigned a speed rating of 100, the minimum speed at nine furlongs is assigned a speed rating of 0. The 100 speed rating line is then drawn for the ordinate 5 with a slope of (-0.0465). Each 0.08 length/sec. at nine furlongs reduces the speed rating by ten until a speed rating of 0 is achieved. This is set forth in the aforementioned patent application and patent.

In evaluating the racing class of each race, the algorithm takes in consideration:

1. Type of race
2. Price, if applicable
3. Purse
4. Sex
5. Age with monthly correctives
6. Class of track

Obviously, horses are flesh and blood and not machines and therefore, the actual races do not exactly match the predicted "speed rating" fro the mathematical algorithm. However, the average fit is extremely close and individual races fit with a standard deviation of +/-5. The difference between the predicted value from the track algorithm and the speed rating of the winner is the race variant. The race variants are then averaged into four categories - "Dirt Sprint", "Dirt Route", "Turf Sprint" and "Turf Route". The separation into dirt and turf is obvious-they are different tracks of different types. The separation into sprint and route is necessary because the clubhouse turn is a standard part of a route trip, but never a part of a sprint trip (bullrings excepted), and the clubhouse turn can have quite different characteristics from the rest of the track.

These averages in four categories are then the correctives (track variants) which are added to or subtracted from the raw speed nubmer in order to correctly answer the question - How fast did the horse run?

The factors which affect the track variant can be divided into two categories, primary and secondary. Primary factors are relatively constant factors such as soil composition and track configuration. Secondary factors are changeable from day to day and even on occasion form race to race. These are factors such as moisture content, cushion depth, tire tracks, wind, etc. It is difficult for the computer to differentiate between statistical clusters and real physical conditions which could cause the track variant to vary race to race.

In fact, tracks typically do not vary much race to race on the same day, but do vary considerably day to day. There are obvious conditions, however, when the track variant does have a significant trend race to race during the day such as cause by rain and snow storms etc.

The track variant which is used to modify the basic formula is as follows:

| y = a + bx | |
|---|---|
| CD S = 10 K<br>SR = 50,0 + 2 x CD | a = 50,0<br>b = 2 |
| 10 K S CD = 50 K<br>SR = 66,25 +0,375 x CD | a = 66,25<br>b =0,375 |
| 50 K S CD s = 100 K<br>SR = 80,0 +0,10 x CD | a = 80,0<br>b =0,10 |
| 100 K S CD<br>SR = 89,44 +0,00556 x CD | a = 89,44<br>b =0,00556 |
| where CD = corrected dollars | |

This formula is then used to generate the graph of Fig. 2 which has Race Rating or speed rating of horses - race rating or the ordinate and corrected dollars on the absicca.

We then take into consideration the track class as follows (all tracks are split in First Class, Second Class and Third Class - The abbreviations are the standard abbreviations for the names of the tracks - AQU is Aqueduct in New York.

1. CORRECTED DOLLARS

1. TRACK CLASS

| a. First Class | b. Second Class | c. Third Class |
|---|---|---|
| AQU | MED | PEN |
| AQI | MTH | COM |
| BEL | ATL | TIM |
| SAR | GS | SUF |
| SA | PHA | EIP |
| HOL | PIM | SPT |
| DMR | BOW | WAS |
| HIA | LRL | FP |
| | DEL | BML |
| | GP | DET |
| | CRC | TDN |
| | CD | RD |
| | KEE | CBY |
| | AP | TRW |
| | LAD | BTC |
| | OP | |
| | FG | |
| | BM | |
| | AKS | |

Then we take into consideration the race types as follows:

2. RACE TYPES

| | | |
|---|---|---|
| A. | OCL $ x 1 | - open claiming |
| b. | BCL $ x .8 | - beaten claiming |
| c. | MCL $ x .5 | - maiden claiming |
| d. | SAL $ x 2 | - starter allowance |
| e. | HCP & STK | |
| | (in thousands) | (corrected dollards) |
| | 500 S = Purse | = 1,000 |
| | 100 S = Purse S500 | = 700 |
| | 50 S = Purse S100 | = 500 |
| | 25 S = Purse S50 | = 100 |
| | Purse S25 | = 75 |

and the allowances:

| f. | ALLOWANCES | | | |
|---|---|---|---|---|
| | | First | Second | Third |
| | MSW | 30 | 20 | 5 |
| | A2 | 35 | 25 | 10 |
| | A3 | 45 | 30 | 15 |
| | A1C | 50 | 30 | 15 |
| | A2C | 60 | 35 | 20 |
| | A3C | 70 | 40 | 25 |
| | A4C | 80 | 45 | 30 |
| | A5C | 90 | 50 | 35 |
| | AT1 | 100 | 50 | 20 |
| | AT2 | 200 | 60 | 25 |
| | AT3 | 300 | 75 | 30 |
| | AT4 | 400 | 80 | 35 |
| | AT5 | 500 | 90 | 40 |

and the universal correctives including sex, age, time of years, and stall in which the horse was bred as follows:

3. UNIVERSAL CORRECTIVES

| 1. | SEX | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MALE<br>Female | Substract 0<br>Add 4 | | | | | | | | | | |
| 2. | AGE AND TIME OF YEAR | | | | | | | | | | | |
| | J | F | M | A | M | J | J | A | S | O | N | D |
| 2 YRS OLD ADD: | 12 | 11 | 10 | 9 | 8 | 8 | 8 | 7 | 6 | 6 | 5 | 5 |
| 3 YRS OLD ADD: | 3 | 3 | 2 | 2 | 2 | 2 | 1 | 1 | 0 | 0 | 0 | 0 |
| 3. | STATE<br>BRED | | | | | | | | | | | |
| | A. | AQI<br>AQU<br>BEL<br>SAR | ADD6 | | | | | | | | | |

All races fall into four categories Sprint - less than 8 furlongs or a route 8 furlongs or more and either dirt or turf.

Also the track is either dirt or turf.

To see how fast a horse ran on a certain day a Hewlett-Packard computer is used with the program set out hereafter to compute the speed rating of the horse in that race on that day. That is the first step, then one corrects that for the track variant. In using the track variant one subtracts the track variant so if the horse ran a certain Speed Rating on a dirt sprint we would subtract the track variant to obtain the corrected speed rating. The program in D-base is as follows:

Now the corrected speed rating of a horse at a certain track at a certain race number at a certain date and a certain race number.

SPEED RATING =

CSR (H,T,D,#) = SR (H,T,D,#) - TV (T,D,C)
H = Horse
T = Track
D = Date
# = Race #
C = Category of Race
Dirt Sprint
Dirt Course
Turf Sprint
Turf Course
These are the Speed Rating Formulas for less than 9 furlongs and greater than 9 furlongs.

$f<9$

$$SR = \frac{\frac{f.60}{t} + (0,077.f) - 4,96}{0,008}$$

$f \geq 9$

$$SR = \frac{\frac{f.60}{t} + (0,0465.f) - 4,686}{0,008}$$

minus the track variant.
N = # Races in that category
i = Running index race # in that category
The corrected race ratings are :

$$TV\ (T,D,C) = \frac{\sum_{i}^{N} [CRR\ i\ (T,D,C) - AL\ (CD \quad (T,D,C)]}{N}$$

Algorithm number is as stated in the graph Fig. 4.
CRR = SR (Winner) + Sex Corrective + State Bred Corrective
CD =
OCL 1 x Price
BCL 0,8 x Price
MCL 0,5 x Price
SAL 2 x Price
SHC 2 x Price
Example:
Race 1 State Bred Filly
SR = 70
Filly + 4 (State Bred) + 6 Corrected Race Rating = 70 + 4 + 6 =80
N = # Races in that category
i = Running index race # in that category

Fig. 3 is a typical chart illustrating the calculations of this invention. We refer to this chart to illustrate a typical application of the track variant. This particular race is at Churchill Downs on May 20, 1987. The first entry is 8 furlongs. Type is OCL (open claiming). The second entry is listed as maiden claiming (MCL). A1C means an allowance race. If it is a claiming race, it means that the horse can be claimed for a preset price.

If it is not a claiming price - there is no price listed (see entries 7 and 8). Every race also has a purse (in thousands of dollars). The claiming price and purse are not the same. In the first case the purse is $7,400, whereas claiming price is $7,500. In the second race the claiming price is $25,000 and the purse is $9,500. The next column lists the sex of the horse; age 2, 3 or older. The next column lists the number of horses in the race - A1 after the age means it is older than the year listed for ex., the first horse is older than 3. The next two columns list Pace Time and the Race Time.

PACE TIME

If the race is less than a mile, the pace is the first half mile. If the race is a mile or over, pace is the first 3/4 mile. Pace is only two distances - half mile or 3/4 mile. Race time is the time of the winner. The next column lists the condition of the track. F means dirt and T means turf-grass. The second letter F or S is fast or slow. It really means whether the track is wet or dry. In an extremely slow track we will see fast. It does whatever it says about the condition of the track has nothing to do with how fast or slow the track is. It has only to do with how wet the track is. So the track may be the slowest track in the world, but it is called fast. It is dry and slow if it is wet.

The next column is corrected dollars. This is an attempt to evaluate the class of all equivalent races at the claiming price. In other words if the race is a claiming race, an open claiming race such as the first listing, the price as the corrected dollars - $7,500 purse will be 7,500 corrected dollars. Maiden claiming races are half the price, because the class of a maiden claiming race is equivalent to half of an open claiming race. So for the second entry the price is $25,000 and the corrected dollars are $12,500. The algorithm includes the aforementioned formulas for generating corrective dollars from all the races. If they are claiming races, open claiming corresponds on a one to one basis, maiden claiming on a 50% basis and then beaten claiming is 80%.

In the allowance races, such as 7 and 8, the starter allowance is multiplied by 2. So if it is 10,000, the starter allowance will be 20,000.00. In the allowance races there is no price. So for those, we use the table alone. All the tracks are put into the first, second and third category (class category).

A2 is a non-winner of two races avenue. A3 is non-winner of three races. A1C is a non-winner of a race other than a maiden or claiming race. A2C is a non-winner of a race other than a maiden or claiming race. A2C is a non-winner of two such races, A3C three such races and so on. This whole allowance classification is non-winners of a dollar amount for a certain time.

PACE RATING (PR):

That is the speed rating formula applied to the pace time on the pace system. We take the speed rating formula and take out F. The pace is always 1/2 mile or 3/4 mile.

SPRINT or ROUTE:

Route is defined as a race of 8 furlongs or over. Thus, 8 furlongs or greater is going to be 6 furlongs at the pace distance. The pace time is always based upon the first 3/4 mile if it is a route race and it is always the first 1/2 mile if it is a sprint race. A sprint is less than eight furlongs and a route is eight furlongs or greater. So a mile or over is a route. With either a 1/2 mile which is 4 furlongs or 3/4 mile, which is six furlongs, we then use the speed rating formula to get the pace rating formula.

To obtain the Pace Variant we take the Pace Rating then we add universal correctives. As stated above universal correctives are sex, age, time of year and where it is bred. For example, if it is a female we add 4, if it is a male, we do not add anything. If the horse is 2 or 3 years or old, you add the amount stated on the table. Also it depends which month we are in, for example. In June for a 2 year old we add 8, 3 year old we add 2. Now, referring to the table - the third entry, it is a 2 year old and the race is in May. So for a 2 year old in May according to the universal correctives we add 8, since the sex of the horse is male. The amount added would be zero. Then computing the PR from the formula we get 51. We then add 8 to get 59. Now we refer to the graph Fig. 4. We look up at the corrected dollars-8.750 and the Race Rating is 67. 67 is what the graph says, 59 is what the race was actually run at. The difference is -8 which is listed as PV. So it is 59 - 67 = -8. This gives us the difference between the projected speed and actual speed. It is the difference between this algorithm from the graph and pace rating with universal correctives.

The graph is derived by looking at hundreds and hundreds of races, and horses at all different prices. What we are doing is comparing this horse with the average horse on a medium fast track for all the male horses. In other words if we have the price and we can tell how fast the race should be run pursuant to this formula. In fact the race in the above example was run slower shows that the track is slow. You could take a great horse, such as Secretariat, and he would run slower on a slow track.

AVERAGE PACE VARIANCE (APV):

To obtain this we take the pace variant in the four categories, viz., route, sprint, dirt, turf. We take an average for each category and place the average in for each race it corresponds to. The break point used for the formulas is different from the definition of sprint or route. So, we use the formula and we compute the number. Some races run fast in the beginning, and some races do not. So this shows you whether they are running fast in the beginning or running fast at the end. We take 57 then add the universal correctives and get -8. If it is an older male, there is no universal correctives. There are no age correctives because it is an older horse. 3-1 means it is older. There are correctives after 3-1. Now let us take the 2 year old (No. 3) we get 43 using the formula. It is a male so there is not any sex correctives, but there is an age corrective. This is May; May for a 2 year old is 8. So we add eight to make 51. The difference is -16. Now, we average them for the 2, 3, 4, 5, 6 and 7 races because they are all the same type of races and the average is -6. Then we put into those races that correspond to it. Now we go to CRR.

Corrected Race Rating (CRR):

We take the race rating and subtract the average race variant. So for the first race 57 - (-4) = 61. This is the normalized rating of the race. Because now we are using the average. So this is the best indication of how fast comparatively the race is really run. Now we go to the next column. This is the corrected race variant. In other words, the race variant -8, but the average race variant was -4 which means that this race was run 4 points slower than what it should have been run. And this race 7 is 10 points faster and race 3 is 10 points slower. So this tells us how fast each race is run versus what it should have been run. This is the normalized condition of the track. This tells you how fast or slow each race was run versus the norm for the day. There is no condition of the track used in this. In other words, this race was 10 points slower after being corrected for the average race variant. This race is still 10 points slower than what it should have been. There are many reasons for it; it could have been not enough competition. It could have been a great horse, but no competition.

While the present invention has been described with reference to preferred embodiments thereof, it is understood by those skilled in the art that various changes in form and application of the electronic speed rating calculator and method may be made without departing from the spirit or scope of the invention.

## Claims

1. An automatic speed rating method for determining the speed rating of an entrant in a speed contest over a predetermined distance comprising, in combination:
determining the world class record for the speed of horses at particular distances
determining the length of the race in furlongs (f)
inputting the above into the relationship

$$SR = \frac{\frac{(f \times 60)}{(t)} + A \times f - B}{0{,}008}$$

modifying such relationships by the track variants for determining the speed rating of the entrant.

2. The automatic speed rating method according to claim 1 wherein the entrants are horses and the race is a horse race over substantially nine furlongs.

3. The automatic speed rating method according to claim 1 wherein for races of 9 furlongs or less A = .077 and B = 4.92 and for races of more than 9 furlongs A = .0465 and B = 4.686.

4. The automatic speed rating method according to claim 3 wherein the track variant is determined in accordance with the formula:

$$TV\ (T,D,C) = \frac{\sum_{i}^{N} [CRRi(T,D,C)-AL(CD(T,D,C)]}{N}$$

Where TV is the track variant.
Where T is time.
Where D is date.
Where C is category.
Where N is number of races in that category.
Where i is the running index of races in that category.
Where CRR is connected race rating.
Where AL is algorithm.
Where CD is connections used on type of race.
Where CRR = SR (Winner) + Sex Corrective + SBC
Where
CD = OCL 1 x Price
BCL .8 x Price
MCL .5 x Price
SAL 2 x Price
SHC 2 x Price

Appendix

```
*** BATCALC - tmp pgm to calc MW speed/variances

clear
set talk off
set safety off
close databases

clear
? 'Indexing RY file'
use ry
if .f.  && ======== to bypass indexing ==========
 set talk on
 index to rr on trak+dtoc(date)+str(race,2)
 set talk off
else
 ? 'bypassed.'
endif

do while .t.
 clear
 gTrk='   '
 use ry index rr
 gMo=month(date())
 @ 5,5 say 'Which reports do you want?'
 @  9,10 say '  Track code: ' get gTrk picture '@!'
 @ 11,10 say 'Month number: ' get gMo picture '##'
 read
 if ''=gTrk
  exit
 endif
 * the next 2 lines change gMo to character, and insure a leading zero
 mm=str(gMo+1000,4)
 mm=substr(mm,3,2)

 use ry index rr
 set exact off
 seek gTrk+mm
 set exact on
 if .not. found()
  @ 10,15 say 'Track or month is invalid'
  wait
  loop
 else
  nxtkey=trak+dtoc(date)+str(race,2)
 endif

 do while substr(nxtkey,1,3)=gTrk .and. substr(nxtkey,4,2)=mm
  use ry index rr
  seek nxtkey
  if .not. found()
   ? '***** Error Condition - "nxtkey" not found'
   ? '***** Call programmer'
   wait
   exit
  endif
  ? 'Pulling data for track/date',gTrk,date,'..  '
  dday=day(date)
  copy to ztrk while trak=gTrk .and. month(date)=gMo .and. day(date)=dday
  nxtkey=trak+dtoc(date)+str(race,2)
  use ztrk
  ?? 'calcs ..  '
  do calc
  ?? 'printing ..'
  do trkrpt
  ?? 'appending to rnew ..'
  use rnew
  appe from ztrk
 enddo

enddo

close databases
set safety on
set talk on
clear
```

```
*** BATCH - print batches of daily reports

* NB - the routines called by this program use many files,
*  so this program must not assume that RY will still be open
*   when a subroutine returns

clear
set talk off
set safety off
close databases

clear
? 'Indexing RY file'
use ry
set talk on
index to rr on trak+dtoc(date)+str(race,2)
set talk off

do while .t.
 clear
 gTrk='   '
 gMo=month(date())
 @  5,5  say 'Which DAILY TRACK reports do you want?'
 @  8,10 say '  Track code: ' get gTrk picture '@!'
 @ 10,10 say 'Month number: ' get gMo picture '##'
 read
 if ''=gTrk
  exit
 endif
 * the next 2 lines change gMo to character, and insure a leading zero
 mm=str(gMo+1000,4)
 mm=substr(mm,3,2)
 * look for the first race of the operator's selection
 use ry index rr
 set exact off
 seek gTrk+mm
 set exact on
 if .not. found()
  @ 15,15 say 'Track or month is invalid'
  wait
  loop
 else
  nxtkey=trak+dtoc(date)+str(race,2)
 endif

 @ 13,0
 do while substr(nxtkey,1,3)=gTrk .and. substr(nxtkey,4,2)=mm
  use ry index rr
  seek nxtkey
  if .not. found()
   ? '***** Error Condition - "nxtkey" not found'
   ? '***** Call programmer'
   wait
   exit
  endif
  ? 'Pulling data for track/date: ',gTrk+' /',date,'..  '
  dday=day(date)
  copy to ztrk while trak=gTrk .and. month(date)=gMo .and. day(date)=dday
  nxtkey=trak+dtoc(date)+str(race,2)
*  ?? 'calcs ..  '
*  do calc
  ?? 'printing ..'
  do trkrpt
 enddo

enddo

close databases
set safety on
set talk on
clear
```

```
*** BKUP - back up

*  parameter Btype = "M" for RM backup, "Y" for RY backup,
*     "RM" for transferring files between computers
parameter btype

close databases
set safety off
set talk off

clear
btype=upper(btype)
do case

case btype='M'
 @ 5,5 say 'Put the RM Backup disk into the A drive'
 ?
 wait
 @ 12,5 say 'Backing up ... '
 run copy rm.dbf a:
 @ 12,0 clear  && clear to end of screen
 @ 12,5 say 'The backup procedure is finished'
 ?
 wait

case btype='Y'
 @ 5,5 say 'Put the RY Backup disk into the A drive'
 @ 7,5 say 'Then respond to the following two prompts by striking any key'
 ?
 run backup c:ry.dbf a:
 @ 12,0 clear  && clear to end of screen
 @ 12,5 say 'The backup procedure is finished'
 ?
 wait

case btype='RM'
 * first check that there really is data in RM; in case operator
 *  accidentally reruns this procedure, it is prevented from
 *   wiping out the good data on the TransferDisk
 use rm
 OK2xfer=.t.
 R=reccount()
 use
 if R=0
  @  9,5 say 'The RM file is empty, so there is nothing to transfer'
  @ 11,5 say '  ( Maybe you have already transferred this data )'
  @ 19,0 say 'This procedure will be aborted'
  @ 20,0
  wait
  OK2xfer=.f.
 endif
 if OK2xfer
  @ 5,5 say 'Put the Transfer disk into the A drive'
  ?
  wait
```

```
   @ 12,5 say 'Transferring to A:ZR ... '
   * create 2 copies on the diskette; the dated one is for security
   run copy rm.dbf a:zr.dbf
   mmdd=substr(dtoc(date()),1,2)+substr(dtoc(date()),4,2)
   @ 13,5 say 'Transferring to A:RM&mmdd. ... '
   run copy rm.dbf a:rm&mmdd..dbf
   * after transfer, zap the RM file
   use rm
   zap
   use
   @ 18,0 clear  && clear to end of screen
   @ 18,5 say 'The transfer procedure is finished'
   ?
   wait
  endif
endcase

close databases
set safety on
set talk on
return
```

```
*** CALC - Calculate speed ratings and variances

set talk off
set safety off
close databases

** open files
select 1
use ztrk
go top
select 2
use rtypes
select 3
use cdage
select 4
use tracks

** determine the track's class
select tracks
 ocate for track=ztrk->trak
cclass=class
select ztrk



*************   Main program loop   ***************
select ztrk
do while .not. eof()

 *** calculate PR and RR
 * PR - pace rating
 tt=str(ptime,6,3)
 pt = 60*int(ptime) + val(substr(tt,4,2)) + .2*val(substr(tt,6,1))
 If dist<8
  replace pr with (240/pt - 4.654) / .008
 else
  replace pr with (360/pt - 4.500) / .008
 endif

 * RR - race rating
 tt=str(rtime,6,3)
 rt = 60*int(rtime) + val(substr(tt,4,2)) + .2*val(substr(tt,6,1))
 if dist<=9
  replace rr with ((60*dist)/rt + .077*dist - 4.962) / .008
 else
  replace rr with ((60*dist)/rt + .0465*dist - 4.686) / .008
 endif

 * find the race type in RTYPES
 select rtypes  && table of race types
 locate for type=ztrk->type
 if found()
  ttype=type
 else
  err=.t.
 ? 'Invalid race type:  ',ztrk->type
endif
select ztrk

*** calculate corrected dollars
do case
case rtypes->cat='C'  && claiming race
 replace cd with price*rtypes->mult
case rtypes->cat='Q'  && quality race
 do case
 case purse >= 500
  replace cd with 1000
 case purse >= 100
  replace cd with 700
 case purse >= 50
  replace cd with 500
 case purse >= 25
  replace cd with 100
 otherwise
  replace cd with 75
 endcase
case rtypes->cat='A'  && allowance race
 do case
 case cclass=1
  replace cd with rtypes->c1cd
 case cclass=2
  replace cd with rtypes->c2cd
 case cclass=3
  replace cd with rtypes->c3cd
 endcase
endcase

*** calculate PV and RV
mmon=month(date)
** first, use track formula to get estimated race rate
do case
case cd<=10
 estrr=50.0  + 2.0*cd
case cd<=50
 estrr=66.25 + 0.375*cd
case cd<=100
 estrr=80.0  + 0.10*cd
case cd>100
 estrr=89.44 + 0.00556*cd
endcase

** PV - pace variant
* initialize
ppv = pr
* sex correction
ppv = ppv + iif(upper(sex)='M',0,4)
* age correction
if age='2 ' .or. age='3 '
 aage=val(age)
 select cdage
```

```
   locate for yrs=aage .and. rcmnth=mmon
   ppv = ppv + corr
   select ztrk
  endif
  * state-bred correction
  if ' '<>sb .and. at(trak,'AQU,AQI,SAR,BEL')=1
   ppv = ppv + 6
  endif
  * finally
  replace pv with ppv-estrr

  ** RV - race variant
  * initialize
  rrv = rr
  * sex correction
  rrv = rrv + iif(upper(sex)='M',0,4)
  * age correction
  if age='2 ' .or. age='3 '
   aage=val(age)
   select cdage
   locate for yrs=aage .and. rcmnth=mmon
   rrv = rrv + corr
   select ztrk
  endif
  * state-bred correction
  if ' '<>sb .and. at(trak,'AQU,AQI,SAR,BEL')=1
   rrv = rrv + 6
  endif
  * finally
  replace rv with rrv-estrr

  *** read next record
  skip
enddo                  &&****** end of main program loop


*** calculate averages ARV and APV
** calculate averages
average pv,rv FOR dist<8 .and.substr(cond,1,1)='T'  to avg1,avg11
average pv,rv FOR dist<8 .and.substr(cond,1,1)<>'T' to avg2,avg12
average pv,rv FOR dist>=8.and.substr(cond,1,1)='T'  to avg3,avg13
average pv,rv FOR dist>=8.and.substr(cond,1,1)<>'T' to avg4,avg14
** insert the averages
replace apv with avg1,arv with avg11;
  for dist<8 .and.substr(cond,1,1)='T'
replace apv with avg2,arv with avg12;
  for dist<8 .and.substr(cond,1,1)<>'T'
replace apv with avg3,arv with avg13;
  for dist>=8.and.substr(cond,1,1)='T'
replace apv with avg4,arv with avg14;
  for dist>=8.and.substr(cond,1,1)<>'T'

*** calculate cpv, cpr, crv, crr, and p_r
replace all cpr with pr-apv, cpv with pv-apv;
```

```
  crr with rr-arv, crv with rv-arv, p_r with pr-rr

close databases
return
```

```
*** CHARTS - controls the data entry cycle

close databases
set talk off
set safety off
clear

lup=.t.
do while lup
 * enter records until operator presses ctrl-W or ctrl-End
 lup2=.t.
 do while lup2
  clear
  use ra
  if reccount()<>0
   @ 5,3 say str(reccount(),5,0)+' races are in the RA file'
   @ 6,0
   wait '         Press any key to continue ... '
  endif
  set format to ra
  append
  kkey=readkey()
  delete all for trak='   ' .and. year(date)=0 .and. race=0
  pack
  if reccount()=0
   lup=.f.
   exit
  endif
  set format to
  use
  clear
  if kkey=12 .or. kkey=270  && Esc/^Q or ^W/^End
   lup2=.f.
  endif
 enddo
 if .not. lup
  exit
 endif
 if kkey=12  && Esc
  lup=.f.
 else
  * verify the entered data
  errflag=.f.
  do verify with errflag
  if errflag
   lup=.t.
   ?
   ? 'Error(s) were found; please fix them.'
   wait
  else
   lup=.f.
   ?
   ? 'The input data has been verified'
   ?
   ?
   ?
   ? 'Now calculating speed ratings and variances ... '
   use ra
   copy to ztrk
   do calc
   ? 'Transferring RA data into RM ... '
   use rm
   append from ztrk
   use ra
   zap
   ? 'Printing the Daily Track Report ... '
   do trkrpt
   ** check for abnormal value of CRV
   use ztrk
   locate for abs(crv)>=15
   if found()
    clear
    @ 5,5 say 'WARNING - one or more races has abnormally high or low'
    @ 6,5 say '           values of CRV'
    @ 10,5 say 'Please inform the office manager, and set aside this'
    @ 11,5 say '  Track Report for his/her inspection'
    @ 16,0
    wait
   endif
  endif  &&errflg
 endif  &&kkey=Esc
enddo

close databases
set talk on
set safety on
return
```

```
*** CONVERT - Convert races files from old to new format

set safety off
set talk off
close databases

clear
ma='A'
@ 5,5 say '[A]rchive or [M]w data: ' get ma picture '!'
read

select 1
use orig alias inp

select 2
use ra
copy stru to conv
use conv
zap

select 3
use rtypes

select 4
use conds

@ 7,0
select inp
do while .not. eof()
 ?? '+'
 select conv
 append blank
 replace trak with upper(ltrim(inp->track))
 replace date with inp->date
 replace race with val(inp->race)
 if ma='A'
  replace dist with inp->distance/100
 else
  replace dist with inp->distance
 endif
 select rtypes
 locate for code=ltrim(inp->type)
 select conv
 if found()
  replace type with rtypes->type
 else
  replace type with upper(ltrim(inp->type))
 endif
 select conv
 if ma='A'
  replace price with inp->price/10
 else
  replace price with inp->price
 endif
 if ma='A'
```

```
   replace purse with inp->purse/10
  else
   replace purse with inp->purse
  endif
  replace sb with upper(inp->stbrd)
  replace sex with upper(inp->sex)
  replace age with ltrim(inp->ag)
  replace hr with inp->hr
  replace pp with inp->pp
  replace odds with inp->wodds
  replace wp with inp->wp
  if ma='A'
   rem = .1*inp->tp - 60*int(inp->tp/600)
   replace ptime with int(inp->tp/600) + .01*int(rem) + .005*(rem-int(rem))
  else
   replace ptime with inp->tp
  endif
  if ma='A'
   rem = .1*inp->tr - 60*int(inp->tr/600)
   replace rtime with int(inp->tr/600) + .01*int(rem) + .005*(rem-int(rem))
  else
   replace rtime with inp->tr
  endif
  select conds
  locate for code=ltrim(inp->condition)
  select conv
  if found()
   replace cond with conds->cond
  else
   replace cond with upper(ltrim(inp->condition))
  endif
  select conv
  replace cd with inp->cd,pr with inp->pr,pv with inp->pv
  replace apv with inp->apv,cpr with inp->cpr,cpv with inp->cpv
  replace rr with inp->rr,rv with inp->rv
  replace arv with inp->arv,crr with inp->crr,crv with inp->crv
  replace p_r with inp->p_r
  select inp
  skip 1
 enddo

 set safety on
 set talk on
 close databases
```

```
*** INIT - FIGS system initialization program

set echo off
set talk off
close databases

set function 9 to [do menu;]
public gDbg
gDbg=.f.

set bell off
set confirm on
set debug off
set delete on
set help off
set print off
set status off
set step off

set talk on
do menu
```

```
*** MENU - main menu program of FIGS system

set talk off
set safety off
close databases
clear

* check that INIT has been run; if not, quit dBASE
if type('gDbg')='U'  && gDbg is undefined
 @ 5,5 say 'F I G S  was not started from DOS properly'
 @ 8,0
 wait '        Press any key to continue ... '
 quit
endif

sel=' '
do while .not. (sel='Q'.or.sel='.')
 sel=' '
 clear
 @ 0,0
 ?'       ======================================================'
 ?'       ||   F.I.G.S. Main Menu                             ||'
 ?'       ||==================================================||'
 ?'       ||                                                  ||'
 ?'       || RM file operations:                              ||'
 ?'       ||   A  -  Enter chart data                         ||'
 ?'       ||   B  -  Make corrections to RM data              ||'
 ?'       ||   C  -  Print monthly reports                    ||'
 if file('figs2')
 ?'       ||   1  -  Move races from RM to TransferDisk       ||'
 endif
 if file('figs1')
 ?'       ||   2  -  Merge TransferDisk races into RM         ||'
 ?'       ||   3  -  Back up RM file                          ||'
 endif
 ?'       ||                                                  ||'
 ?'       || RY file operations:                              ||'
 if file('figs1')
 ?'       ||   4  -  Move races from RM to TransferDisk       ||'
 endif
 if file('figs2')
 ?'       ||   5  -  Merge TransferDisk races into RY         ||'
 ?'       ||   6  -  Back up RY file                          ||'
 endif
 ?'       ||                                                  ||'
 ?'       || - - - - - - - - - - - - - - - - - - - - - - - -  ||'
 ?'       ||   U  -  Utilities menu                           || '
 ?'       ||   Q  -  Quit      .  -  dBASE                    ||'
 ?'       ||                                                  ||  '
 ?'       ======================================================   '
 ?
 ?
 rr=row()
 @ 24,7 say '=== F.I.G.S. copyright (C) 1987, Robert Sinn  ==='
 @ rr,7 say '   Select one of the above: ' get sel picture '!'
 read
 @ 22,0
 xx=' '
 do case
 case file('figs1').and.at(sel,'156')<>0
  @ 22,10 say 'Invalid selection on this machine.  Press Enter ' get xx
  read
  if readkey()<>270
   sel=' '
  endif
 case file('figs2').and.at(sel,'234')<>0
  @ 22,10 say 'Invalid selection on this machine.  Press Enter ' get xx
  read
  if readkey()<>270
   sel=' '
  endif
 endcase
 do case
 case sel='Q' .or. sel='.'
  exit
 case sel='U'
  do menuU
 case sel='A'
  do charts
 case sel='B'
  do rfix
 case sel='C'
  do mtd
 case sel='1'
  do bkup with 'RM'
 case sel='2'
  do merge with 'M'
 case sel='3'
  do bkup with 'M'  && RM backup
 case sel='4'
  do xtrmon
 case sel='5'
  do merge with 'Y'
 case sel='6'
  do bkup with 'Y'  && RY backup
 endcase
 set talk off
enddo

close databases
set safety on
set talk on
if sel='Q'
 quit
else
 clear
 return
endif
```

```
*** MENUU - FIGS utility menu

set talk off
set safety off
close databases
clear

sel=' '
do while sel<>'E'
 sel=' '
 clear
 @ 2,0
 ?'              ----------------------------------------------'
 ?'              ¦    F.I.G.S. Utility Menu                   ¦'
 ?'              ¦--------------------------------------------¦'
 ?'              ¦                                            ¦'
 ?'              ¦    S  -  Speed ratings calculator          ¦'
 ?'              ¦    C  -  Count the races in a file         ¦'
 if file('figs2')
 ?'              ¦    M  -  Monthly reports from RY           ¦'
 ?"              ¦    F  -  Full month's track reports from RY¦"
 endif
 ?'              ¦                                            ¦'
 ?'              ¦    E  -  Exit to main menu                 ¦'
 ?'              ¦    Q  -  Quit     .  -  dBASE              ¦'
 ?'              ¦                                            ¦'
 ?'              ----------------------------------------------'
 ?
 ?
 @ row(),10 say 'Select one of the above: ' get sel picture '!'
 read
 @ 22,0
 xx=' '
 if file('figs1').and.at(sel,'MF')<>0
  @ 22,10 say 'Invalid selection on this machine.  Press Enter ' get xx
  read
  if readkey()<>270
   sel=' '
  endif
 endif
 do case
 case sel='E' .or. sel='Q' .or. sel='.'
  exit
 case sel='C'
  do rcount
 case sel='F'
  do batch
 case sel='M'
  do mo
 case sel='S'
  do sr
 endcase
 set talk off
enddo
```

```
close databases
set talk on
set safety on
return
```

```
*** MERGE - merge races from diskette file into RM or RY

* rtype='M' to merge into RM
* rtype='Y' to merge into RY
parameter rtype
rtype=upper(rtype)

close databases
set talk off
set safety off

* copy the diskette file into a work file on the hard disk
clear
@ 5,5 say 'Put the Transfer disk into the A drive'
wait
@ 5,0 clear
@ 5,5 say 'Transferring TransferDisk races to C drive ... '
if rtype='M'
 run copy a:zr.dbf c:
else  && rtype='Y'
 run copy a:zr2.dbf c:zr.dbf
endif

* check against duplicate races
@ 8,5 say 'Checking TransferDisk races for doubles ... '
select 1
if rtype='M'
 use rm alias rr
else  && rtype='Y'
 use ry alias rr
endif
index on trak+dtoc(date)+str(race,2) to rr
select 2
use zr
dups=0
do while .not. eof()
 kkey=trak+dtoc(date)+str(race,2)
 select rr
 seek kkey
 if found()
  dups=dups+1
  if dups=1
   clear
   @ 0,5 say 'The following "new" races are already in R'+rtype+':'
   ?
  endif
  ? '       Track',trak,'    Date',date,'    Race',str(race,3)
 endif
 select zr
 skip 1
enddo

* don't continue if duplicates were found
if dups=0
 OK2merge=.t.
```

```
else
 OK2merge=.f.
 ?
 ? '         Double-entured races were found'
 ? '         You must correct the problem before running'
 ? '           this procedure again'
 ?
 wait
endif

* if no dups, merge the work file into RM or RY
if OK2merge
 @ 11,5 say 'Merging TransferDisk races ... '
 close databases
 if rtype='M'
  use rm alias rr
 else  && rtype='Y'
  use ry alias rr
 endif
 append from zr
endif

close databases
set talk on
set safety on
```

```
*** MO - print monthly reports off of RY

* NB - the routines called by this program use many files,
*  so this program must not assume that RY will still be open
*   when a subroutine returns

clear
set talk off
set safety off
close databases

* guard against absence of RY
if .not. file('RY.DBF')
 @ 5,5 say 'File RY is not on this computer'
 ?
 wait
 set talk on
 set safety on
 return
endif

? 'Indexing the RY file ..'
use ry
set talk on
index to ry on trak+dtoc(date)
set talk off

do while .t.
 clear
 gTrk='   '
 gMo=0
 @ 5,5 say 'Which MONTHLY report do you want?'
 @ 7,10 say '  Track code: ' get gTrk picture '@!'
 @ 8,10 say 'Month number: ' get gMo picture '##'
 read
 if ''=trim(gTrk) .or. readkey()=12
  exit
 endif
 * the next 2 lines change gMo to character, and insure a leading zero
 mm=str(gMo+1000,4)
 mm=substr(mm,3,2)
 * look for the first race of the operator's selection
 use ry index ry
 set exact off
 seek gTrk+mm
 set exact on
 if .not. found()
  @ 10,15 say 'Track or month is invalid'
  wait
  loop
 endif
 copy to zmon while trak=gTrk .and. month(date)=gMo
 gEnd=32  && cutoff day-of-month does not apply to full-month reports
 do morpt
enddo
```

```
close databases
set safety on
set talk on
clear
```

```
*** MORPT - print monthly report

set talk off
set safety off
close databases

* get some values used later
use zmon
* month number, and first of month date..
mmon=month(date)
day1=date
do while day(day1)<>1
 day1=day1-1
enddo
* name of track..
ttrak=trak
use tracks
locate for track=ttrak
if found()
 nmtrk=upper(trkname)
else
 nmtrk=upper(ttrak)
endif

zsep =replicate(' .',67)
zsep2=replicate('  ',67)
use zmon
set print on
store space(132) to h1,h2,h3,h4

? chr(18)+space(50)+nmtrk
? space(20)+'TRACK VARIANTS'
? space(20)+'--------------'+space(16)+'Month: '+cmonth(date)
?
?
? space(26)+'D I R T'+space(23)+'T U R F'
? chr(015)
spc=space(5)
sp1=space(4)
sp2=space(42)
sp3=space(16)
ds1='   (DSV)     '
ds2='Dirt Sprint '
dr1='   (DRV)     '
dr2='Dirt Route   '
ts1='   (TSV)     '
ts2='Turf Sprint '
tr1='   (TRV)     '
tr2='Turf Route   '
var='  Variant    '
und='-------------'
? sp2+ds1+sp1+dr1+sp3+'           '+ts1+sp1+tr1
? sp2+ds2+sp1+dr2+sp3+'           '+ts2+sp1+tr2
? '            DAY                   COND        '+var+sp1+var;
  +sp3+'COND       '+var+sp1+var
```

```
? '           ---                   ----       '+und+sp1+und;
  +sp3+'----       '+und+sp1+und
goto top
dday=day1
do while month(dday)=mmon .and. day(dday)<=gEnd
* ? dday,day(dday),gend
* wait
 if date<>dday .or. eof()
  ? str(day(dday),13,0)+'   -dark-'
 else
  store 0 to ds,dsn,dr,drn,ts,tsn,tr,trn
  store '    ' to condD,condT
  do while date=dday
   cnd=substr(cond,1,1)
   do case
   case dist<8.and.cnd<>'T'
    ds=ds+arv
    dsn=dsn+1
    condD=cond
   case dist>=8.and.cnd<>'T'
    dr=dr+arv
    drn=drn+1
    condD=cond
   case dist<8.and.cnd='T'
    ts=ts+arv
    tsn=tsn+1
    condT=cond
   case dist>=8.and.cnd='T'
    tr=tr+arv
    trn=trn+1
    condT=cond
   endcase
   skip 1
  enddo
  store '    none    ' to dsa,dra,tsa,tra
  if dsn>0
   dsa = str(ds/dsn,4,0)+'   ('+str(dsn,2,0)+' )'
  endif
  if drn>0
   dra = str(dr/drn,4,0)+'   ('+str(drn,2,0)+' )'
  endif
  if tsn>0
   tsa = str(ts/tsn,4,0)+'   ('+str(tsn,2,0)+' )'
  endif
  if trn>0
   tra = str(tr/trn,4,0)+'   ('+str(trn,2,0)+' )'
  endif
  ? str(day(dday),13,0) +space(20) +condD +spc +dsa +sp1 +dra;
                         +space(18) +condT +spc +tsa +sp1 +tra
 endif
 if dow(dday)=5
  ? zsep2
 endif
 if dow(dday)=1
  ? zsep
 endif
 dday=dday+1
enddo

?
? CHR(018)
? '    Copyright  1987  Robert Sinn                                               Re
v A'
set print off
eject

close databases
return
```

```
*** MTD - print month-to-date Monthly reports

* NB - the routines called by this program use many files,
*  so this program must not assume that RM will still be open
*   when a subroutine returns

clear
set talk off
set safety off
close databases

* prompt for month and cutoff-day, after calculating defaults for them
clear
gMo=month(date())
d=date()
do while month(d)=gMo
 d=d+1
enddo
gEnd=day(d-1)
@ 5,5 say 'Print reports for month ' get gMo  picture '##' range 1,12
@ 7,5 say '            through day ' get gEnd picture '##' range 1,31
@ 20,8 say 'Esc - Cancel this procedure'
read
kk=readkey()
if kk=12
 set talk on
 set safety on
 return
endif

@ 20,0 clear
? 'Indexing the RM file ..'
use rm
set talk on
index to rm on trak+dtoc(date)
set talk off

do while .t.
 clear
 @ 5,5 say 'Print reports for month ' + str(gMo,2)
 @ 7,5 say '            through day ' + str(gEnd,2)
 gTrk='   '
 @ 9,5 say '              for track ' get gTrk picture '@!'
 read
 if ''=trim(gTrk) .or. readkey()=12
  exit
 endif
 * the next 2 lines change gMo to character, and insure a leading zero
 mm=str(gMo+1000,4)
 mm=substr(mm,3,2)
 * look for the first race of the operator's selection
 use rm index rm
 set exact off
 seek gTrk+mm
 set exact on
 if .not. found()
  @ 11,15 say 'Track or month is invalid'
  wait
  loop
 endif
 copy to zmon while trak=gTrk .and. month(date)=gMo
 do morpt
enddo

close databases
set safety on
set talk on
clear
```

```
*** RCOUNT - Count the races in a file: by track; by day or month

set talk off
set safety off
close databases

clear
@ 5,5 say 'Track-Date Contents Report'
dm='D'
ff=space(20)
@ 8,5 say 'Which file: ' get ff picture '@!'
@ 10,5 say 'Count by [D]ay or by [M]onth: ' get dm picture '!'
read
ff=trim(ff)

* quit program if file name was not entered
OK=(''<>ff)

* check that file exists
if OK .and. .not. file(ff+'.dbf')
 OK=.f.
 @ 15,5 say 'File '+ff+' is not in this directory'
 ?
 wait
endif

* check for empty file
if OK
 use &ff
 if reccount()=0
  OK=.f.
  @ 15,5 say 'File '+trim(ff)+' is empty'
  ?
  wait
 endif
endif

if OK
 use &ff
 @ 15,5 say 'Sorting ..'
 set talk on
 sort to zr on trak,date,race
 set talk off
 @ 15,0 clear to 16,79
 use zr
 cc=0
 xtr=trak
 xda=date
 xmon=month(xda)
 mgn=space(10)
 set print on
 ? mgn,date,'      Track/Date Contents Report'
 ?
 mgn=mgn+space(7)
 if dm='D'
```

```
  ? mgn,'Track    Date          Races'
 else
  ? mgn,'Track    Month        Races'
 endif
 ?
 do while .not. eof()
  if trak<>xtr .or. iif(dm='D',date<>xda,month(date)<>xmon)
   cmo=cmonth(xda)+space(10-len(cmonth(xda)))
   ? mgn,xtr,'    ',iif(dm='D',dtoc(xda),cmo),str(cc,6)
   cc=0
   xtr=trak
   xda=date
   xmon=month(xda)
  endif
  cc=cc+1
  skip 1
 enddo
 cmo=cmonth(xda)+space(10-len(cmonth(xda)))
 ? mgn,xtr,'    ',iif(dm='D',dtoc(xda),cmo),str(cc,6)
 ?
 ? mgn,'*** End of Report ***'
 set print off
 eject
endif

set talk on
set safety on
close databases
return
```

```
*** RFIX - extract all races for one track/day into RA

set talk off
set safety off
close databases
qOK=.t.

clear
use ra
if reccount()<>0
 @ 5,5 say 'The RA file is not empty'
 @ 6,5 say '  Please correct the situation, then try this procedure again'
 @ 7,5
 wait '         Press any key to continue ... '
 qOK=.f.
endif

if qOK
 append blank
 @ 5, 5 say 'Pull data from RM into RA for:'
 @ 7,15 say '    track: ' get trak picture '@!'
 @ 8,15 say 'race date: ' get date
 read
 xtrak=trak
 xdate=date
 if ''=xtrak
  qOK=.f.
 endif
endif

if qOK
 use rm
 copy to ra for trak=xtrak.and.date=xdate
 use ra
 nRecs=reccount()
 if nRecs=0
  @ 12,5 say 'No races were found for the specified track and date'
  wait '         Press any key to continue ... '
 else
  use rm
  delete for trak=xtrak.and.date=xdate
  pack
  do charts
 endif
endif

close databases
set safety on
set talk on
return
```

```
*** SR - program to calculate speed ratings

close databases
set talk off
set safety off
f=0
t=0
l=0
do while .t.
 clear
 @ 3,5  say 'F.I.G.S. dBASE Speed Rating Calculator'
 @ 6,10 say '      Furlongs:   ' get f picture '##.##'
 @ 8,10 say '  Time (M.SSF):   ' get t picture '#.###'
 @ 10,10 say 'Lengths behind:   ' get l picture '##.##'
 @ 14,22 say 'Press Esc to quit'
 read
 if f=0 .or. readkey()=12
  exit
 endif
 tt=str(t,6,3)
 rt = 60*int(t) + val(substr(tt,4,2)) + .2*val(substr(tt,6,1))
 if f<=9
  sr = ((60*f-1)/rt + .077*f - 4.962) / .008
 else
  sr = ((60*f-1)/rt + .0465*f - 4.686) / .008
 endif
 @ 14,0
 @ 14,22 say 'Speed Rating =' + transform(sr,'####.#')
 @ 20,0
 wait
enddo

close databases
set talk on
set safety on
return
```

```
*** TRKRPT - Daily Track Report

close databases
set talk off
select 1
use ztrk

select 2
use tracks
locate for track=ztrk->trak
if found()
 ttrack=trkname
else
 ttrack=ztrk->trak
endif
ttrack=trim(upper(ttrack))

select 1
goto top
index on race to ztrk
SET PRINT ON
? CHR(015)
? ' '
? ' '
? space(50)+' T R A C K   R E P O R T '
?
? space(50)+space((25-len(ttrack))/2)+ttrack
?
? space(50)+'   Racedate: ',date
? ' '
go top
?' #  Dist Type  Price  Purse  Sex SB Age HR PP   Odds WP  Ptime  Rtime ';
  +' Cond   Cor$   PR   PV  APV  CPR  CPV   RR   RV  ARV  CRR  CRV  P_R'
DO WHILE .NOT. eof()
 ? race,dist,type,str(price,7,1)+str(purse,7,1),'';
   ,sex,' ',sb,'',age,str(hr,3),pp,odds,wp;
   ,str(ptime,6,3),str(rtime,6,3),'',cond,str(cd,7,1);
   ,pr,pv,apv,cpr,cpv,rr,rv,arv,crr,crv,p_r
 SKIP
ENDDO
?' '
?'  Copyright  1987  Robert Sinn'+space(95)+'Rev A'
?CHR(018)
SET PRINT OFF
eject
close databases
return
```

```
*** VERIFY - checks the races input data

parameter err

set talk off
set safety off
clear
close databases

select 1
use ra
select 2
use rtypes
select 3
use tracks
select 4
use conds

** check to be sure the RA file contains only one day's races
**  on one track
select ra
goto top
tt=trak
dd=date
locate for trak<>tt .or. date<>dd
if found()
  err=.t.
  ? '*** RA file problem'
  ? '   All races must be for the same track and date, but more than one'
  ? '     track or date was found in this RA file'
endif


***** main loop - each loop is one race
select ra
goto top
DO While .not. eof()
 qq1=.t.

 * delete any record with date,track, and race missing
 if year(date)=0.and.trak='    '.and.race=0
  delete
  skip 1
  loop
 endif

 * before verifying data, convert any numeric codes that were entered
 *  for track condition and race type into their alpha equivalents
 *   Also, left adjust alfa fields
 * track
 select ra
 replace trak with ltrim(trak)
 * race type
 select rtypes
 locate for code=ltrim(ra->type)
```

```
if found()
 replace ra->type with type
endif
* age
select ra
replace age with ltrim(age)
* condition
select conds
locate for code=ltrim(ra->cond)
if found()
 replace ra->cond with cond
endif
select ra


** track
select tracks
locate for track=ra->trak
if .not.found()
 err=.t.
 if qq1
  ? '*** Race'+str(ra->race,3,0)+':'
  qq1=.f.
 endif
 ? '   Invalid track:  ',ra->trak
endif
select ra

** check race type and figure out category
ccat=' '
select rtypes  && table of race types
locate for type=ra->type
if found()
 ccat=cat
else
 err=.t.
 if qq1
  ? '*** Race'+str(ra->race,3,0)+':'
  qq1=.f.
 endif
 ? '   Invalid race type:  ',ra->type
endif
select ra

** claiming-race price
if ccat='C' .and. price=0
 err=.t.
 if qq1
  ? '*** Race'+str(ra->race,3,0)+':'
  qq1=.f.
 endif
 ? '   Claiming race, but price is missing'
endif

** non-claiming-race price
```

```
if ccat<>'C' .and. price<>0
 err=.t.
 if qq1
  ? '*** Race'+str(ra->race,3,0)+':'
  qq1=.f.
 endif
 ? '   Non-claiming race, price is not zero'
endif

** purse
if purse=0
 err=.t.
 if qq1
  ? '*** Race'+str(ra->race,3,0)+':'
  qq1=.f.
 endif
 ? '   Purse is missing'
endif

** state-bred
if at(sb,' Y')=0
 err=.t.
 if qq1
  ? '*** Race'+str(ra->race,3,0)+':'
  qq1=.f.
 endif
 ? '   State-bred code invalid:  ',sb
endif

** sex
if at(sex,'MF')=0
 err=.t.
 if qq1
  ? '*** Race'+str(ra->race,3,0)+':'
  qq1=.f.
 endif
 ? '   Sex code invalid:  ',sex
endif

** age
if at(age,'2 ,3 ,34,31,4 ,41')=0
 err=.t.
 if qq1
  ? '*** Race'+str(ra->race,3,0)+':'
  qq1=.f.
 endif
 ? '   Age code invalid:  ',age
endif

** HR - number of horses
if hr=11 .or. hr=22
 if qq1
  ? '*** Race'+str(ra->race,3,0)+':'
  qq1=.f.
 endif
```

```
  ? '   HR is 11 or 22 - possibly due to pressing key too long'
  aa=' '
  accept '     Type "A" to Accept it anyway: ' to aa
  if upper(aa)<>'A'
   err=.t.
  endif
 endif

** PP - post position
if pp>hr
 err=.t.
 if qq1
  ? '*** Race'+str(ra->race,3,0)+':'
  qq1=.f.
 endif
 ? '   PP is greater than HR'
endif
if pp<=hr .and. (pp=11 .or. pp=22)
 if qq1
  ? '*** Race'+str(ra->race,3,0)+':'
  qq1=.f.
 endif
 ? '   PP is 11 or 22 - possibly due to pressing key too long'
 aa=' '
 accept '     Type "A" to Accept it anyway: ' to aa
 if upper(aa)<>'A'
  err=.t.
 endif
endif

** WP - winning position
if wp>hr
 err=.t.
 if qq1
  ? '*** Race'+str(ra->race,3,0)+':'
  qq1=.f.
 endif
 ? '   WP is greater than HR'
endif
if wp<=hr .and. (wp=11 .or. wp=22)
 if qq1
  ? '*** Race'+str(ra->race,3,0)+':'
  qq1=.f.
 endif
 ? '   WP is 11 or 22 - possibly due to pressing key too long'
 aa=' '
 accept '     Type "A" to Accept it anyway: ' to aa
 if upper(aa)<>'A'
  err=.t.
 endif
endif

** PR - pace rating
tt=str(ptime,6,3)
pt = 60*int(ptime) + val(substr(tt,4,2)) + .2*val(substr(tt,6,1))
```

```
 If dist<9
  vpr = (240/pt - 4.654) / .008
 else
  vpr = (360/pt - 4.500) / .008
 endif
 if vpr<30 .or. vpr>99
  if qq1
   ? '*** Race'+str(ra->race,3,0)+':'
   qq1=.f.
  endif
  ? '   PR is abnormally low or high:',vpr,'      ( PTime =',ptime,')'
  aa=' '
  accept '     Type "A" to Accept it anyway: ' to aa
  if upper(aa)<>'A'
   err=.t.
  endif
 endif

 ** RR - race rating
 tt=str(rtime,6,3)
 rt = 60*int(rtime) + val(substr(tt,4,2)) + .2*val(substr(tt,6,1))
 if dist<=9
  vrr = ((60*dist)/rt + .077*dist - 4.962) / .008
 else
  vrr = ((60*dist)/rt + .0465*dist - 4.686) / .008
 endif
 if vrr<30 .or. vrr>99
  if qq1
   ? '*** Race'+str(ra->race,3,0)+':'
   qq1=.f.
  endif
  ? '   RR is abnormally low or high:',vrr,'      ( RTime =',rtime,')'
  aa=' '
  accept '     Type "A" to Accept it anyway: ' to aa
  if upper(aa)<>'A'
   err=.t.
  endif
 endif

 *** read next record
 SKIP
ENDDO

***** end of main program loop

close databases
return
```

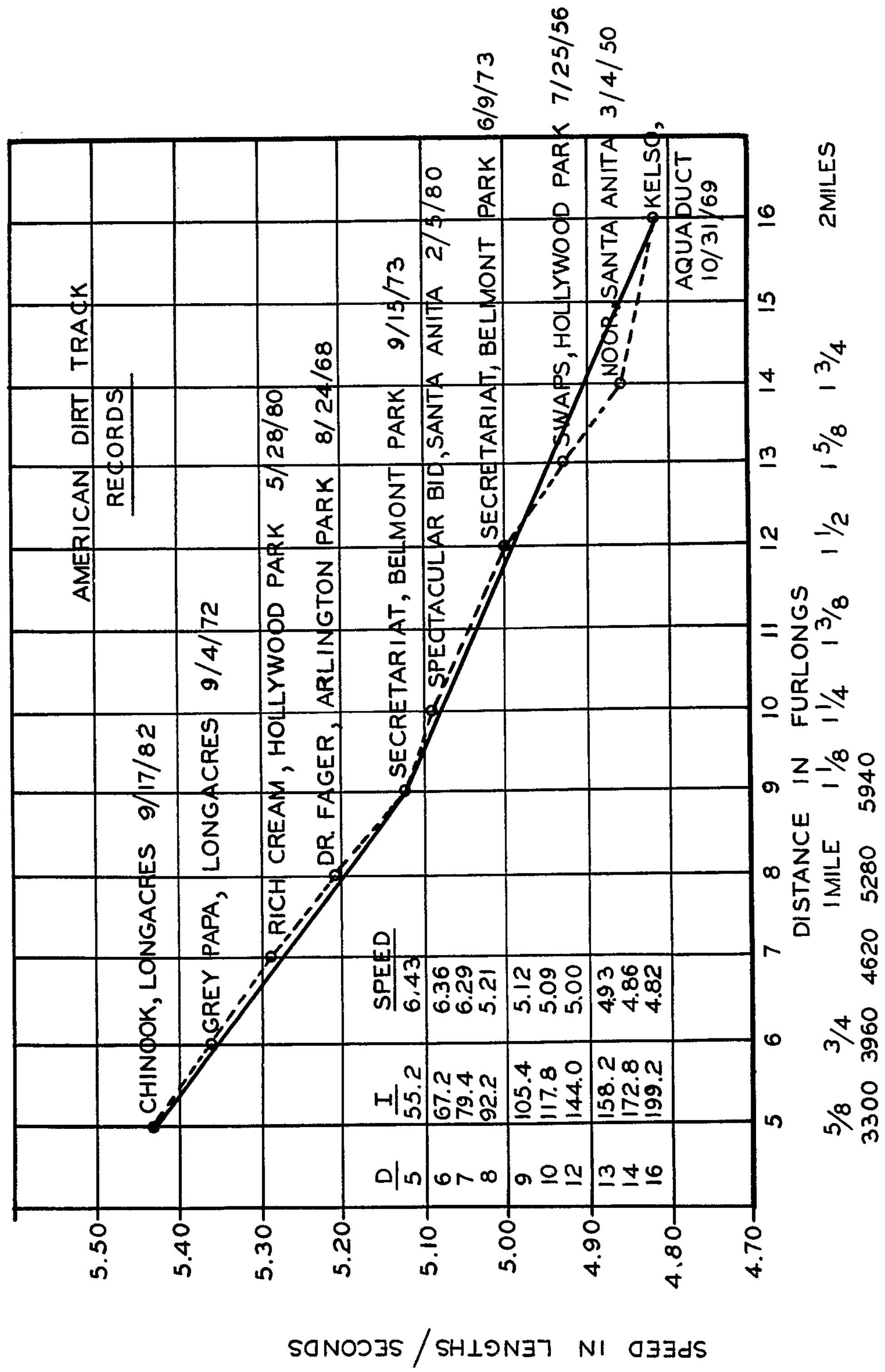
```
*** XTRMON - extract one month's charts from RM

set talk off
set safety off
close databases
clear

gMo=0
@ 5,5 say 'Enter the number of the month you want to transfer ';
  get gMo picture '##'
read
OKxtr=.t.
if gMo=0 .or. readkey()=12
 OKxtr=.f.
endif
if OKxtr
 dd=ctod(str(gMo)+'/1/99')  && any date with the selected month
 @ 7,5 say 'About to transfer '+cmonth(dd)+' races to diskette'
 @ 8,5 say '  If OK, press Enter;  if not OK, press Esc  '
 read
 if readkey()=12
  OKxtr=.f.
 endif
endif
if OKxtr
 * copy the records to a:zr2
 @ 12,5 say 'Put the Transfer disk into the A drive'
 ?
 wait
 @ 14,0
 @ 14,5 say 'Copying races to file A:ZR2 on diskette ... '
 use rm
 copy to a:zr2 for month(date)=gMo
 @ 15,5 say 'Deleting those races from the disk ... '
 delete for month(date)=gMo
 pack
endif

close databases
set safety on
set talk on
```

FIG. 1


AMERICAN DIRT TRACK
RECORDS
CHINOOK, LONGACRES 9/17/82
GREY PAPA, LONGACRES 9/4/72
RICH CREAM, HOLLYWOOD PARK 5/28/80
DR. FAGER, ARLINGTON PARK 8/24/68
SECRETARIAT, BELMONT PARK 9/15/73
SPECTACULAR BID, SANTA ANITA 2/3/80
SECRETARIAT, BELMONT PARK 6/9/73
SWAPS, HOLLYWOOD PARK 7/25/56
NOOR, SANTA ANITA 3/4/50
KELSO, AQUADUCT 10/31/69
D I SPEED
5 55.2 6.43
6 67.2 6.36
7 79.4 6.29
8 92.2 5.21
9 105.4 5.12
10 117.8 5.09
12 144.0 5.00
13 158.2 4.93
14 172.8 4.86
16 199.2 4.82
5.50
5.40
5.30
5.20
5.10
5.00
4.90
4.80
4.70
SPEED IN LENGTHS / SECONDS
5 6 7 8 9 10 11 12 13 14 15 16
DISTANCE IN FURLONGS
5/8 3/4 1 MILE 1 1/8 1 1/4 1 3/8 1 1/2 1 5/8 1 3/4 2 MILES
3300 3960 4620 5280 5940

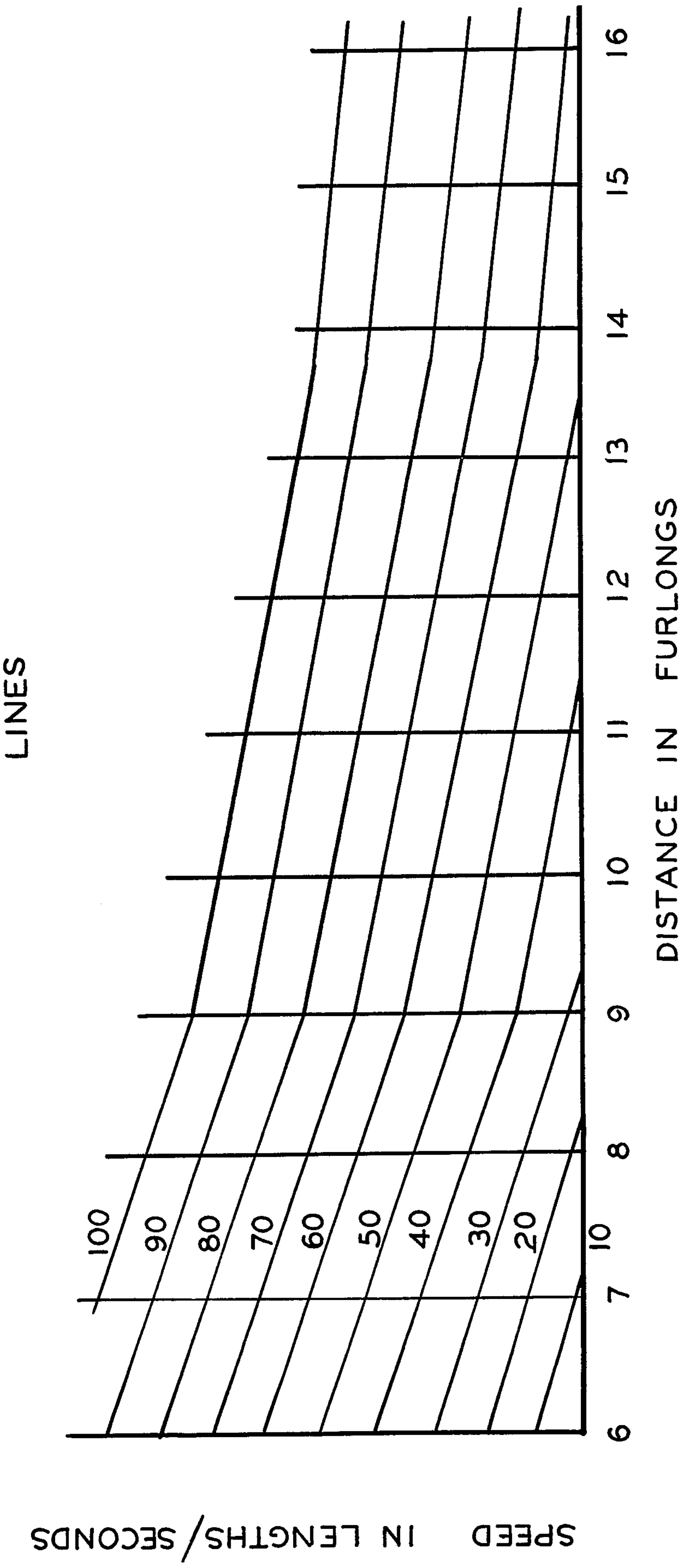


FIG. 2
SPEED RATING LINES

Fig. 3

TRACK REPORT

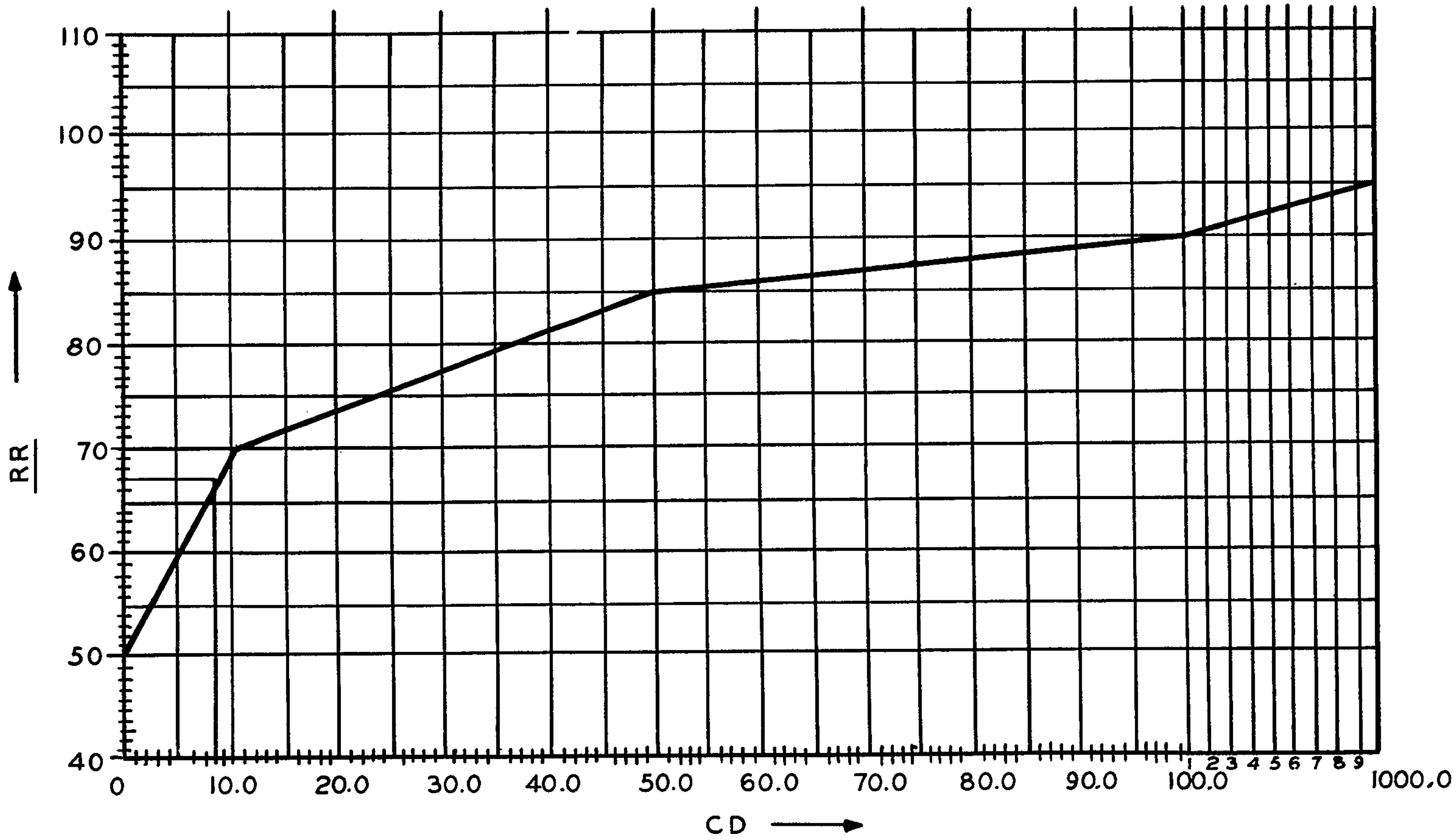
CHURCHILL DOWNS

Racedate: 05/20/87

| # | Dist | Type | Price | Purse | Sex | Age | Rtime | Cd |
|---|---|---|---|---|---|---|---|---|
| 1 | 8.50 | OCL | 7.500 | 7.400 | M | 31 | 1.470 | 7.500 |
| 2 | 6.00 | MCL | 25.000 | 9.500 | F | 31 | 1.122 | 12.500 |
| 3 | 5.00 | MCL | 17.500 | 8.000 | M | 2 | 1.010 | 8.750 |
| 4 | 8.00 | OCL | 10.000 | 7.000 | F | 31 | 1.393 | 10.000 |
| 5 | 6.00 | OCL | 25.000 | 11.000 | F | 3 | 1.120 | 25.000 |
| 6 | 6.00 | OCL | 10.000 | 8.500 | M | 41 | 1.112 | 10.000 |
| 7 | 6.00 | A1C | 0.000 | 21.000 | M | 31 | 1.094 | 30.000 |
| 8 | 8.00 | A1C | 0.000 | 20.000 | F | 31 | 1.383 | 30.000 |
| 9 | 8.00 | OCL | 17.500 | 11.000 | M | 41 | 1.364 | 17.500 |

| PR | PV | APV | CPR | CPV | RR | RV | ARV | CRR | CRV | P R |
|---|---|---|---|---|---|---|---|---|---|---|
| 44 | -21 | -12 | 56 | -9 | 57 | -8 | -4 | 61 | -4 | -13 |
| 70 | 3 | 1 | 69 | 2 | 59 | -8 | -4 | 65 | -2 | 11 |
| 51 | -8 | 1 | 50 | -9 | 43 | -16 | -6 | 49 | -10 | 8 |
| 54 | -12 | -12 | 66 | 0 | 59 | -7 | -4 | 63 | -3 | -5 |
| 73 | 3 | 1 | 72 | 2 | 62 | -8 | -6 | 68 | -2 | 11 |
| 76 | 6 | 1 | 75 | 5 | 68 | -2 | -6 | 74 | 4 | 3 |
| 79 | 2 | 1 | 78 | 1 | 82 | 4 | -6 | 88 | 10 | -3 |
| 57 | -16 | -16 | 73 | 0 | 65 | -8 | -8 | 73 | 0 | -8 |
| 71 | -2 | -12 | 83 | 10 | 77 | 4 | -4 | 81 | 8 | -6 |

FIG. 4


RR
110
100
90
80
70
60
50
40
0
10.0
20.0
30.0
40.0
50.0
60.0
70.0
80.0
90.0
100.0
1
2
3
4
5
6
7
8
9
1000.0
CD

European Patent Office

# EUROPEAN SEARCH REPORT

Application Number

EP 89 10 7924

## DOCUMENTS CONSIDERED TO BE RELEVANT

| Category | Citation of document with indication, where appropriate, of relevant passages | Relevant to claim | CLASSIFICATION OF THE APPLICATION (Int. Cl.5) |
|---|---|---|---|
| D,A | US-A-4 133 031 (ESRAC)<br>* Column 2, line 40 - column 3, line 53 *<br>--- | 1,3 | G 06 F 15/28 |
| A | US-A-4 382 280 (MATTEL)<br>* Column 1, lines 27-51 *<br>--- | 1,3 | |
| A | GB-A-2 093 237 (DUTCHFORD)<br>* Page 1, line 4 - line 84; figures 1-4 *<br>--- | 1,3 | |
| A | FR-A-2 515 389 (DUBARRY)<br>* Page 1, line 19 - page 2, line 23; page 2, line 39 - page 3, line 14; page 4, line 37 - page 6, line 36; figure 2 *<br>----- | 1,3 | |
| | | | TECHNICAL FIELDS SEARCHED (Int. Cl.5) |
| | | | G 06 F 15/28 |

The present search report has been drawn up for all claims

| Place of search | Date of completion of the search | Examiner |
|---|---|---|
| THE HAGUE | 05-12-1989 | CHUGG D.J. |

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document