A remote control vehicle is presented that has a telemetry sensor, which detects wheel rotation, and a track sensor that reads track indicators found on a track over which the vehicle is driven. The vehicle receives commands from a mobile device to control its operation. Wheel rotation data and track sensor data are transmitted from the vehicle back to the mobile device for storage along with the commands that were transmitted to the vehicle. The commands, rotation data, and track sensor data are transmitted to a server computer over a wide area network, and thereafter shared with another user as run data. When downloaded by the other user, the run data can be used to compare the other user’s ability to control their vehicle with the run data. The run data can further be used to control the other user’s vehicle.
Fig. 1

100

110
Servers

112
Data

120
Network

130

132

140

142

150

152
Fig. 2

Mobile Device

- Display
- User Input
- Processor
- Local Wireless Interface
- Network Interface

Memory
- Operating System
- Remote Control App
- User Controls / Input Data
- Car Telemetry Data
- Track Related Data
- Image Data

Network Interface

Server

- Processor
- Network Interface

Memory

Operating System
Application Programming

Database

- Users
- Vehicles
- Runs
- Images and Video
- Telemetry Data
- Track Data

Network
Fig. 4

Infrared Sensor 400

Infrared Transmitting Tube 410

Infrared Receiving Tube 430

Gear 420

Logic 310
Start

Lay out Track with Track Indicators

Establish Link Between Vehicle and Mobile Device

Place Vehicle on Track

Use Control App on Mobile Device to Traverse Track While Storing Data

Acquire All Data for Run at Mobile Device

Upload Run Data to Cloud Server including Control Inputs

Download Third-Party Run Data from Another User from Cloud Server

Compare Third-Party Run Data with User's Prior Run

Use Control App for New Run, With Direct Comparison with Third-Party Run

User Third-Party Run Data To Control Car

End
Receive Car Control input on Mobile Device

Transmit Car Control input to Vehicle

Receive Car Control input from Mobile Device

Adjust Car Performance to Reflect Car Control Input

Read (Record) Data from Telemetry Sensors

Read (Record) Data from Track Sensors

Read (Record) Data from Image Sensors

Receive, Display, and Store Data from Vehicle Along with Control Inputs

Transmit Data to Mobile Device

End
Fig. 9

900 Start

905 Receive Command to Set Controlled Start Mode

910 Transmit Mode Change

915 Set Mode Change in Vehicle

920 Receive Starting Throttle Command

925 Transmit Throttle Command

930 Start Motor at Maximum Throttle

935 Sense Movement Through Track Sensors

940 Reduce Throttle and Prevent Throttle From Exceeding Limit Set Below Max Throttle

End
TOY VEHICLE WITH TELEMETRICS AND TRACK SYSTEM AND METHOD

CROSS REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit of U.S. Provisional Application Ser. No. 61/885,440, filed Jul. 25, 2013, which is hereby incorporated by reference in its entirety herein.

FIELD OF INVENTION

[0002] The present application relates to the field of toy vehicles. More particularly, the described embodiments relate to a remote control vehicle with on-board telemetry systems that monitor performance and a remote control application that monitors inputs by a user for storage and sharing over a cloud-based computer network.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] FIG. 1 is a schematic diagram showing a system for controlling remote control vehicles and for recording and sharing of data relating to the vehicles.
[0004] FIG. 2 is a schematic diagram showing details for a mobile device, a server, and a related database for one embodiment of the system of FIG. 1.
[0005] FIG. 3 is a schematic diagram showing details for a remote controlled vehicle and track for one embodiment of the system of FIG. 1.
[0006] FIG. 4 is a front plan view of a detailed section of the gearing for a remote controlled car with an infrared RPM telemetry sensor.
[0007] FIG. 5a is a schematic diagram showing a first embodiment of a track having a visual track indicator.
[0008] FIG. 5b is a schematic diagram showing a second embodiment of a track having a visual track indicator.
[0009] FIG. 5c is a schematic diagram showing a third embodiment of a track having a visual track indicator.
[0010] FIG. 6 is a front plan view of a tablet computer showing an embodiment for a user interface for use with the system of FIG. 1.
[0011] FIG. 7 is a flow chart showing a method for operating the system of FIG. 1.
[0012] FIG. 8 is a flow chart showing a method for detecting and transmitting data in connection with the method of FIG. 7.
[0013] FIG. 9 is a flow chart showing a method for controlling the start of the vehicle.

DETAILED DESCRIPTION

System 100

[0014] FIG. 1 shows a system 100 that utilizes one or more cloud-based servers 110 to store data in a database 112 related to the operation of one or more remote controlled vehicles 140, 142. Vehicle 140, for instance, is controlled by a first tablet computer 130. An app (not shown in FIG. 1) running on the tablet computer 130 allows a user to input commands to the vehicle 140. The vehicle 140, in turn, utilizes telemetry sensors to record data about its operation, such as its speed, the rotations per minute (RPM) of one or more wheels, acceleration, distance travelled, etc. These sensors are read using digital logic found on the vehicle 140. The data from these sensors is then transmitted to the tablet computer 130 for storage and analysis using the app operating on the tablet computer 130. In some embodiments, the remote controlled vehicle 140 operates on a track 150. In these embodiments, the track 150 can contain indicators that can be read by track sensors found on the vehicle 140. The track sensors can be, for instance, image sensors that read data, lines, and other markings found on the track. The information read by the track sensors can also be transmitted by the remote controlled vehicle 140 to the tablet computer 130 for analysis and storage.

[0015] The tablet computer 130 stores the telemetry and track-related data that it receives from the vehicle 140 in its memory. This data can be transferred through a wide-area network 120 (such as the Internet) to one or more remote servers 110 for storage in the remote database 112. In addition, the tablet computer 130 can record the input commands that it received from the user and store these inputs along with the telemetry and track data in its memory for transmission to the remote server 110. In one embodiment, this information is grouped together for each race (or “run”) through the track by the first vehicle 140. A run is a particular race by the vehicle 140 through the track 150, and may constitute one or more laps of the track 150. In other embodiments, the vehicle 140 need not operate on a particular track 150, and a run is a set time period during which the remote controlled vehicle 140 is operated and data is collected and stored.

[0016] A second user using a second tablet computer 132 can access the input commands, telemetric data, and track data that the first user stored in the database 112. This information can be used by the second tablet computer 132 to compare the first user’s performance with car 140 to the second user’s ability to control car 142. In particular, the second user can lay out a track 152 that is identical to the track 150 used by the first user. Since the second tablet computer 132 has downloaded all of the information about the first user’s run on their track 150, the second user can use their tablet computer 132 to control their vehicle 142 through the same track layout 152 and compare the results. In effect, the two physical vehicles 140, 142 are permitted to race head-to-head through the same track layout 150, 152. This is true even though the second vehicle 142 may make its run through the track 152 at a later time and at a different location than the runs made by the first vehicle 140 through its track 150.

Mobile Device 210 and Server 260

[0017] FIG. 2 shows a remote control vehicle 200 that is in communication with a mobile device 210. In particular, the vehicle is communicating with a local wireless interface 218 found on the mobile device 210. This interface may take the form of a BLUETOOTH® connection that complies with one of the standards of the Bluetooth Special Interest Group (such as the Bluetooth 4.0). Alternatively, the interface 218 may be a Wi-Fi interface that utilizes one of the Institute of Electrical and Electronics Engineers’ (IEEE) 802.11 standards.

[0018] The mobile device 210 can take the form of a smartphone or tablet computer. As such, the device 210 will include a display 212 for displaying information to a user, a user input mechanism 214 for receiving user input commands (such as a touch screen integrated into the display 212), and a processor 216 for processing instructions and data for the device 100. The display 212 can use LCDs, OLEDs, or similar technology to provide a color display for the user. The processor 216 can be a general purpose CPU, such as those provided by Intel Corporation (Mountain View, Calif.) or Advanced Micro...
The mobile device 210 also has the local wireless interface 218 for interfacing with the remote control vehicle 200, and a wide area network interface 220 for connecting with a network 250. In some embodiments, these interfaces 218, 220 could be the same interface. For instance, the mobile device 210 may interface with both the remote controlled vehicle 200 and the network 250 over a single Wi-Fi network interface 218, 220. The mobile device 210 further includes a memory 230 for storing processing instructions and data. This memory is typically solid-state memory, such as flash memory. Mobile devices such as device 210 generally use specific operating systems 232 designed for such devices, such as iOS from Apple Inc. (Cupertino, Calif.) or ANDROID OS from Google Inc. (Menlo Park, Calif.). The operating system 232 is stored on the memory 230 and is used by the processor 216 to provide a user interface for the display 212, to receive input over the user input device(s) 214, to handle communications for the device 210 over the interfaces 218, 220, and to manage applications (or apps) that are stored in the memory 230, such as the remote control app 234.

The remote control app 234 is responsible for receiving user input 214 related to the control of the remote controlled vehicle 200 and ensuring that these inputs are relayed to the vehicle 200 over interface 218. In addition, the app 234 receives data from the vehicle 200 over interface 218 and stores this data in memory 230. In particular, the app 234 may receive car telemetry data 238 and track related data 240. In addition, some embodiments of the app 234 allow the user to request that the vehicle 200 take video or still images using an image sensor found on the vehicle 200. This image data 242 is also received by the app 234 and stored in memory 230. In addition to storing this data 238, 240, 242, the app 234 also generates a user interface on the display 212 and shares this data in real-time with the user over the display 212. Finally, the app 234 is responsible for connecting with a remote server 260 over network interface 220 and sharing its data 238, 240, 242 with the server 260. The app 234 can also request data from the server 260 concerning previous runs or runs made by third-parties, and store this data in memory 230.

The server 260 contains a programmable digital processor 262, such as a general purpose CPU manufactured by Intel Corporation (Mountain View, Calif.) or Advanced Micro Devices, Inc. (Sunnyvale, Calif.). The server 260 further contains a wireless or wired network interface 264 to communicate with remote computing devices, such as mobile device 210, over the network 250. The processor 262 is programmed using a set of software instructions stored on a non-volatile, non-transitory, computer readable memory 266, such as a hard drive or flash memory device. The software typically includes operating system software 268, such as LINUX (available from multiple companies under open source licensing terms) or WINDOWS (available from Microsoft Corporation of Redmond, Wash.).

The processor 262 controls the communication with mobile device 210 under the direction of an application program 269 residing on the server memory 266. The application program 269 is further responsible for maintaining data within a database 270. The database 270 may be stored in the memory 266 of the server 260 as structured data (such as separate tables in a relational database, or as database objects in an object-oriented database environment). Alternatively, the database 270 may be stored and managed in a separate device, such as a separate database server computer and/or a storage area network storage device. Database programming stored on the memory 266 directs the processor 262 to access, manipulate, update, and report on the data in the database 270. This database programming can be considered part of the application programming 269.

FIG. 2 shows the database 270 with tables or objects for users 272, vehicles 274, races or runs 276, user inputs 278, telemetry data 280, track data 282, and image data 284. Relationships between the database entities are represented in FIG. 2 using crow’s foot notation. For example, FIG. 2 shows that each user 272 can be associated with one or more vehicles 274, since each actual user may have multiple remote control vehicles 200. In addition, a vehicle 200 may be shared by more than one user, so the database 270 allows a particular vehicle database entity 274 to be associated with multiple user database entities 272. A particular run of the vehicle 200 over a track is represented by a single run database entity 276, which is associated with a single user 272 and a single vehicle 274. For each run 276, the database 270 can store and track multiple user inputs 278, telemetry data readings 280, track data readings 282, and images and/or video files 284. Associations or relationships between the database entities shown in FIG. 2 can be implemented through a variety of known database techniques, such as through the use of foreign key fields and associative tables in a relational database model. In FIG. 2, associations are shown directly between two database entities, but entities can also be associated through a third database entity. For example, an image file 284 is directly associated with one run 276, and through that relationship the image file 284 is also associated with a single vehicle 274 and a single user 272.

Each of the database entities 272-284 shown in FIG. 2 can be implemented as a separate relational database table within a relational database 260. Alternatively, each of the entities 272-284 could be implemented using a plurality of related database tables. It is also possible to implement the entities 272-284 as objects in an object oriented database. The distinction made between the entities 272-284 in FIG. 2 and the following description are made for ease in understanding the data maintained and manipulated by the server 260, and should not be seen to limit the scope of the invention described herein.

Mobile Device 210 and Server 260

FIG. 3 schematically reveals the major electrical components of a remote controlled vehicle 200. The vehicle 200 has a processor 310 that controls the major functions of the vehicle 200. In the embodiment shown in FIG. 3, the processor 310 operates under the control of operational programming 322, which is stored in digital memory 320 that is accessible by the processor 310. Alternatively, the processor 310 and the operational programming 322 could be combined into an application-specific integrated circuit (or “ASIC”) or a field programmable device (such as an FPGA) specifically designed to handle the processing requirements of the vehicle 200. Whether formed as a general-purpose processor, an ASIC, or an FPGA, the processor 310 receives instructions for operating the vehicle from a tablet computer 210 over a local wireless interface 318. As described above, this interface 318 may operate under Bluetooth protocols, IEEE 802.11 protocols, or any similar local, wireless protocols. Based on these control instructions, the operational programming
will control the wheel motor(s) 330 and the steering motor 332 to control forward and backward motion and the steering of the vehicle. 200.

[0026] The instructions received from the tablet computer 210 may include directions for the vehicle 200 to take still or video images on its image sensor(s) 316. In some embodiments, the resulting images 324 are stored in the vehicle memory 320 for transmission back to the tablet computer 210 when convenient. In the preferred embodiment, the data from the image sensors 316 is fed to the mobile device 210 as a live feed, allowing the tablet computer 210 to generate still and video image files as desired by the user.

[0027] While the vehicle 200 is moving under control of these instructions, the processor 310 is monitoring the telemetry sensors 312 to obtain data about how the vehicle 200 is moving and behaving. These sensors 312 may include RPM sensors that track wheel revolutions for the vehicle, accelerometer sensors that track acceleration of the vehicle, and even sensors that measure the performance and characteristics (such as heat) of the wheel motors 330. Preferably, the telemetry sensors 312 include at least an RPM sensor that can indicate wheel rotations, from which can be derived the vehicle speed and distance traveled. In some embodiments, separate RPM sensors 312 may be placed on wheels driven by the wheel motors 330 and non-driven wheels. The sensors 312 at the powered wheels may detect wheel spin during periods of hard acceleration, at the time sensors 312 at the non-driven wheels will give a better indication of the vehicle’s current speed and distance traveled.

[0028] FIG. 4 shows one embodiment of a telemetry sensor 400. In particular, sensor 400 is an infrared RPM sensor that measures rotations of a wheel on the vehicle 200. This sensor 400 utilizes an infrared transmitting tube 410 that transmits an infrared beam through a gear 420 that is used to drive a wheel of the vehicle 200. On the other side of the gear 420 is an infrared receiving tube 430 that receives and senses the infrared beam transmitted by transmitting tube 410. When the gear 420 rotates, a portion of the gear 420 will interrupt the beam during the rotation. By counting the interruptions in the infrared signal detected by the infrared receiving tube 430, the processor/logic 310 can determine rotations of the wheel on the vehicle 200. Each interruption may not indicate a complete wheel rotation, as the gear 420 will likely interrupt the infrared signal multiple times in a single rotation, and a single rotation of the gear 420 will not likely lead to a single rotation of the wheel. Nonetheless, based on the gearing of the vehicle and the construction of the gear 420, there will be a known relationship between interruptions in the light beam and rotations of the wheel.

[0029] In addition to monitoring telemetry sensors 312 such as the infrared RPM sensor 400 shown in FIG. 4, the processor 310 also monitors one or more track sensors 314 on the vehicle 200 (as seen in FIG. 3). These sensors 314 read track indicators 352 found on a specially designed track 350. In the preferred embodiment, the track 350 constitutes a plurality of specially designed track pieces of a plurality of known lengths that can be configured into multiple track layouts. These track segments are constructed with track indicators 352 that can be read by the track sensor 314 on the vehicle 200. As explained below in connection with FIGS. 5a-5c, these track indicators 352 can be visual markings on the surface of the track 350. In these cases, the track sensor 314 takes the form of an image sensor 314 that, together with the processor 310, can recognize and interpret the visual markings 352. In other embodiments, the track indicators 352 can take the form of radio frequency identifies, such as specially designed RFID tags, that can be read while the vehicle 200 passes over the track 350 by a reading sensor 314 on the vehicle 200.

Track

[0030] As explained above, in one embodiment the track sensor 314 is an image sensor that can detect visual markings 352 found on the track 350. FIGS. 5a-5c show three example markings that may be placed on the track 350. In FIG. 5a, two track segments 510, 520 of differing lengths are shown. Each track segment 510, 520 contains alternating background colors 512, 514, with a lighter background 512 always being followed by a darker background 514. Each segment 512, 514 is of a uniform width. The image sensor 314 is pointed downward on the vehicle 200 toward the track segments 510, 520. When the sensor 314 notes a change in background color on the track 512, 514, the processor 310 will know that the vehicle 200 has traveled the distance necessary to move to the next background segment 512, 514. When this information is transmitted to the mobile device 210, the device 210 will be able to verify that the vehicle 200 was indeed travelling along the track 350. Furthermore, this track traversal information can be compared with distance information obtained from the telemetry sensor 312 to ensure that all of the measurements are consistent with vehicle 200 movement on the track 350.

This prevents competitors from spoofing the system 100, such as by faking a run through a track while holding a vehicle 200 in the air.

[0031] FIG. 5b shows another technique for creating visual track markings 352. In this Figure, the track segments 530, 540 each have color lines 532, 534, 536 perpendicular to the length of the track segment at regularly spaced intervals. The sensors 314 can read the presence and color of these lines 532-536 and transmit this information to the processor 310. Using this information, the tablet computer 210 can also determine distance traversal along the track 350. By alternating three colors, the tablet computer 210 can identify situations where a vehicle has reversed directions and is traversing backwards along the track 350. The use of three different identifiable shades or colors could also be used in the embodiment shown in FIG. 5a to detect backwards movement along the track.

[0032] In FIG. 5c, the track segments 550, 560 reach have a visual bar code 552, 562 that can be read as the track sensor 314 passes along the track 350. Each code 552, 562 identifies the track segment 550, 560, which can be used to determine the length and configuration of that track segment 550, 560. In one embodiment, the codes 552, 562 are identification numbers, and the length and configuration of the related segments 550, 560 must be determined by consulting a look-up table located on the vehicle memory 320, the tablet computer 210, or on the remote server (110, 260). In other configurations, this information is directly read from the codes 552, 562 that are printed on the track segments 550, 560.

[0033] Regardless of which marking system is used, it is possible to create special markings on the track segments 510-560 to indicate unique locations on the track 350. For instance, a finish line could be created with a unique background color in track segment 510, or with a unique color line in segment 530, or with a unique code in segment 550. In this way, the vehicle 200 can sense when it crosses the finish line,
in order to stop timing a particular run, or to count laps when a particular race involves multiple laps of the track 350.

User Interface

[0034] FIG. 6 shows a user interface 600 on a mobile device 210 that is generated by the remote control app 234. The main portion of the interface 600 shows a live view 602 of the image being seen by the image sensor 316 on the vehicle 200. In this case, the vehicle 200 is seen following a hiker on a dirt trail. The interface 600 allows the user to control the movement of the controlled vehicle through direction controls 610 and gear controls 620. The direction controls 610 are shown as four directional arrows superimposed on the image 602. As the tablet computer 210 uses a touch screen for user input 214, the user need only touch one of the four direction arrows 610 to change the direction or steer the vehicle 200. The gear controls 620 are composed of an up and down arrow superimposed over the image 602, allowing the user to change gears by simply pressing one of the arrows 620. Speed and gear information 630 are shown on the bottom of the interface 600.

[0035] At the top of the interface are a variety of controls. The touch mode control changes the steering controls from the directional controls 610 to tilt control. Tilt control uses sensors within the mobile device 210 to determine when a user tilts the device 210 and sends signals to steer the vehicle in the direction of the tilt. The timer display at the center top of the interface 600 allows a user to see the current time of a particular “run.” The photo button stores the current image as a still image, while the video button causes the mobile device 210 to record the image stream coming from the vehicle 200 as a video file. In FIG. 6, a red circle is placed on the video button to indicate that a video file is currently being created of the image feed being displayed at location 602. The help button presents a help system to the user, while the screen control menu allows the user to change various settings with the interface 600 and the app 234 as a whole.

Method

[0036] FIG. 7 shows a method 700 for utilization of the vehicle 200, mobile device 210, server 260, and track 350. The method starts at step 705 with the arrangement of track segments, such as segments 530, 540, into a track 350. The track 350 may be arranged in a complex loop, where the vehicle 200 can traverse the closed loop multiple times for a single race. This is not necessary, however, as the track 350 may be arranged with a separate beginning and an end.

[0037] At step 710, a wireless communication path is established between the vehicle 200 and the mobile device 210. As explained above, this can be a Bluetooth connection, a Wi-Fi connection, or any similar wireless data communication path between the devices 200, 210. At step 715, the vehicle 200 is placed on the track 350. At step 720, the mobile device 210 is used to control the vehicle in a race or run around the track 350. At the same time (or immediately after the race), the mobile device acquires data related to the race (step 725).

[0038] These steps 720, 725 are described in more detail in connection with flow chart 800 in FIG. 8. In particular, the mobile device 210 receives car control input from the user at step 805, and then transmits control signals reflecting that input to the vehicle 200 in step 810. The vehicle 200 receives those signals at step 815, and then adjusts the car performance in step 820. In particular, the vehicle can manipulate the wheel motors 330 and the steering motor 332 to control motion and direction of the vehicle 200. Other control inputs and responses are possible depending on the capabilities of the vehicle 200. For example, the vehicle may have additional motors to control such things as a camera mount, a crane, a plow blade, or additional drive wheels; or the vehicle may have additional electronic components such as a microphone, multiple cameras, a speaker, a touch sensor, etc. The user interface 600 allows a user to control such features, and control inputs for these features can be transmitted at step 810, received at step 815, and implemented at step 820.

[0039] While the vehicle 200 is in motion, the telemetry sensors 312, the track sensors 314, and the image sensors 316 will be read at steps 825, 830, and 835 respectively. In one embodiment, the data from these sensors 312-316 will be immediately transmitted to the mobile device 210 at step 840. The mobile device 210 will receive this data at step 845, and then display and/or store the data. In other embodiments, the data from the sensors 312-316 will be stored in the memory 320 of the vehicle 200 for transmission to the mobile device 210 at a later time. In the preferred embodiment, all of this data will be time coded so as to be able to compare each element of data temporally with the other data, including the received control signals.

[0040] Returning to step 730 of FIG. 7, the mobile device 210 will upload the data that it acquires for a particular run of the vehicle to the cloud server 260 for storage in the database 270. This includes not only the telemetry data 326 and the track data 328, but also the user inputs 236 that were used to drive the vehicle 200. In some embodiments, even the image data is uploaded to the server 260 and stored in the database 270.

[0041] At step 735, the mobile device 210 downloads from the server 260 data from a race run by a third party. This allows the user of the mobile device 210 to compare the third-party race results with their own. If the third party ran the race on the same track configuration, it would be possible to compare the performance of each user head-to-head. The total time for the race could be compared to determine a race winner. Individual statistics could be compared, such as fastest lap time, longest wheel skid, etc. If the user elects to perform their race again at step 745, the third-party race results could be displayed live on the interface 600 while the user controlled their vehicle 200 over the track 350. The interface 600 could display the user’s lead over the opponent, or how far behind their opponent their vehicle 200 is at any point in the race. The interface 600 could even superimpose an image of the third-party vehicle on the image portion 602 of the interface 600 whenever the user was running behind the third party during their race.

[0042] At step 750, the third-party user input data 236 could even be used to control the user’s vehicle 200. While environmental differences and variations in starting positions may prevent input data 236 from a multi-lap third-party race from recreating the entire race with the user’s vehicle, this ability could prove useful to recreate maneuvers performed by third parties. For example, a third-party may be able to perform interesting tricks with their vehicle based on a particular sequence of input commands. The third-party could upload the run data for the trick. When the current user sees video of the trick (such as on a website operated using the data in database 270), they decide to download the third-party run
data for the trick. At step 745, they use the third-party user inputs to control their own physical vehicle 200 as the trick is recreated in front of them.

[0043] FIG. 9 shows an additional method 900 that uses the sensors 312, 314 on the vehicle 200 to create a controlled start mode. One problem with racing toy vehicles 200 is that their motors 330 are usually engineered to provide a great deal of torque to the vehicle wheels relative to the vehicle’s size and weight. Noveice users tend to run their first race at “full throttle” and soon find the vehicle 200 impossible to control. When they attempt their next race at a lower throttle, they find that the vehicle 200 is difficult to start quickly from a stopped position without sufficient torque being supplied by the wheel motor 330. Method 900 solves this issue by implementing a controlled start mode that may useful for novice users.

[0044] The method 900 starts at step 905, in which the mobile device 210 receives a command through the user interface to put the vehicle 200 into a controlled start mode. At step 910, the mobile device 210 transmits this mode change to the vehicle 200. At step 915, the vehicle 200 receives this transmission and sets the vehicle 200 to operate in this mode.

[0045] Some time latter, the mobile device 210 receives a command through its user interface to increase the throttle of the vehicle 200 to a user-selected level (step 920). This command is then sent as an instruction to the vehicle 200 at step 925. The vehicle 200 receives this command at step 930. However, instead of operating the motor 330 at the user-selected level, the vehicle 200 recognizes that it is not currently moving and has been set to operate in the controlled start mode. As a result, the vehicle starts the motor 330 operating at maximum torque (aka “throttle”) in step 930. This allows the vehicle 200 to start as quickly as possible.

[0046] At step 935, the vehicle 200 senses that it has begun to move. This sensing can be accomplished through the track sensor 314. In other embodiments, the telemetry sensor can determine that the vehicle wheels are now moving the vehicle 200. Once the vehicle 200 has begun to move, the vehicle automatically reduces the amount of torque being provided from the motor 330 to the vehicle wheels (step 940). In one embodiment, the amount torque being provided by the motor 330 at step 940 is completely controlled by the throttle commands received from the mobile device 210. If the user is request 90% throttle, step 940 will now provide 90% throttle. In another embodiment, the user throttle commands are not allowed to drive the motor 330 above a pre-set throttle/torque level, where such level is below the maximum level provided at step 930. For example, to help novice users control the vehicle, the controlled start mode may prevent the motor from ever providing greater than 60% throttle (other than during start-up at step 930). If the user requests 20% throttle, the motor 330 is operated at 20%, but if the user requests 100% throttle, the motor 330 is operated at the pre-set level (such as 60%).

[0047] In another embodiment, the vehicle 200 always operates its motor 330 as instructed by the mobile device 210. In this embodiment, the mobile device 210 becomes responsible for operating the vehicle 200 in controlled start mode. To accomplish this, the vehicle 200 must constantly provide telemetry data 326 and track data 328 to the mobile device 210. In this way, the mobile device 210 will know when the vehicle 200 is stopped and when it begins moving. The mobile device 210 will instruct the vehicle 200 to operate the motor 330 at full throttle when starting, as described above in connection with step 930. When the data received from the vehicle 200 indicates to the mobile device 210 that the vehicle 200 has started moving, the mobile device 210 will change its throttle command to the vehicle 200 to operate under controlled throttle (as described above in connection with step 940).

[0048] The many features and advantages of the invention are apparent from the above description. Numerous modifications and variations will readily occur to those skilled in the art. Since such modifications are possible, the invention is not to be limited to the exact construction and operation illustrated and described. Rather, the present invention should be limited only by the following claims.

What is claimed is:
1. A remote control vehicle system comprising:
   a) a remote control vehicle having:
      i) a wireless vehicle communication interface for receiving commands and transmitting data,
      ii) wheels,
      iii) a motor to control wheels in response to the received commands
      iv) a wheel sensor to create wheel rotation data, and
      v) a vehicle processor to transmit wheel rotation data across the wireless vehicle communication interface
   b) a mobile computing device having:
      i) at least one wireless mobile device communication interface for
         (1) transmitting commands to the vehicle,
         (2) receiving wheel rotation data from the vehicle,
         (3) communicating with a remote server computer
      ii) a user interface to receive commands to be transmitted to the vehicle,
      iii) mobile device memory to store commands and received wheel rotation data, and
      iv) a mobile device processor to manage commands and received wheel rotation data and to direct transmission of commands and data to the remote server computer;
   c) the remote server computer having:
      i) a server communication interface for receiving commands and wheel rotation data from the mobile computing device,
      ii) a server memory containing a database of structured data, and
      iii) a server processor for receiving commands and wheel rotation data over the server communication interface and for storing the input commands and wheel rotation data in the database.
2. The vehicle system of claim 1, wherein the mobile device processor receives third-party commands from the remote server computer and transmits the third-party commands to the remote control vehicle.
3. The vehicle system of claim 2, wherein the third-party commands were received by the remote server from a third-party mobile computer device that previously used the third-party commands to drive a third-party vehicle.
4. The vehicle system of claim 1, further comprising:
   d) track have track indicators,
   wherein the remote control vehicle further comprises a track sensor that reads the track indicators off the track as the vehicle passes over the track.
5. The vehicle system of claim 1, wherein the vehicle processor operates in a controlled start mode when the vehicle starts from a stopped position, wherein the torque provided by
the motor to the wheels is limited to a predetermined value when the vehicle begins to move.

6. The vehicle system of claim 5, wherein the torque provided by the motor to the wheels is maximized when the vehicle is in the stopped position.

7. The vehicle system of claim 5, wherein the wheel sensor is used to determine whether or not the vehicle is moving.

8. The vehicle system of claim 5, wherein the mobile device controls the torque provided by the motor during the controlled start mode.

9. The vehicle system of claim 5, wherein the vehicle controls the torque provided by the motor during the controlled start mode.

10. A method for operating a remote control vehicle comprising:
    a) laying out track having track indicators;
    b) locating the remote control vehicle on the track;
    c) controlling the remote control vehicle by inputting commands on a user interface on a mobile device, wherein the mobile device sends commands to the remote control vehicle;
    d) on the remote control vehicle, reading track data by sensing the track indicators found on the track using a track sensor on the vehicle as the vehicle passes over the track;
    e) sending track data from the remote control vehicle to the mobile device;
    f) sending track data and the commands from the mobile device to a remote server.

11. The method of claim 10, wherein the remote control vehicle accumulates wheel rotation data and sends wheel rotation data to the mobile device.

12. The method of claim 11, wherein the wheel rotation data is sent with the track data and commands from the mobile device to the remote server.

* * * * *