May 18, 2015

Automatic Selection of Noise Abatement Parameters For Touch Controllers

Milton Ribeiro

Automatic Selection of Noise Abatement Parameters For Touch Controllers

Abstract: When a power adapter is connected to a computing device (e.g., a notebook, a laptop, a tablet, a smartphone, etc.), the power adapter identifies itself by sending identifier information such as, for example, manufacturer, part number, revision, manufacturing date, etc. The computing device queries a locally-stored table of known power adapters to obtain related touch controller data (e.g., list of hopping frequencies, frequency hopping strategy, filtering strategy and parameters, etc.) to be used by a touch controller in the computing device for noise abatement. If the power adapter is not on the locally-stored table, the computing device queries a remotely-stored table (e.g., table stored in a cloud service), to obtain the touch controller data. The touch controller data is provided to the touch controller, which uses the data to configure a noise abatement strategy. An estimate of the load on the power adapter is provided periodically to the touch controller for as long as the power adapter is connected to the computing device.

In a computing device having a touch input device (e.g., a touchscreen and/or a touchpad), noise interference may be experienced when a power adapter is connected to the computing device in combination with a user’s finger on the touch input device. For example, charger or finger-coupled noise is the interference introduced by common-mode power adapter noise that is coupled onto the touch input device (e.g., touchscreen and/or touchpad) by the user's finger. That problem has been made very relevant by the abundance of low cost, high noise, micro-USB (Universal Serial Bus) power adapters in the market. The problem is also expected to be a problem in type-C USB systems.

Traditionally, abatement of this type of noise is performed through techniques deployed by the manufacturer of the touch controller. For example, the techniques include techniques such as frequency hopping, linear and nonlinear filtering, multiple-frame filtering and others. The employed techniques can be made more effective if the particulars of the power adapter in question are known so that adequate controller parameters can be chosen. If the particulars of the power adapter are not known, default dynamic algorithms can provide good results but many times the noise is not abated sufficiently and the result is a poor user experience.

To address the shortfalls of the traditional noise abatement solution, the following system and techniques are proposed. When a power adapter is connected to a computing device (e.g., a notebook, a laptop, a tablet, a smartphone, etc.), the power adapter identifies itself by sending identifier information such as, for example, manufacturer, part number, revision, manufacturing date, etc. The computing device queries a locally-stored table of known power adapters to obtain related touch controller data (e.g., list of hopping frequencies, frequency hopping strategy, filtering strategy and parameters, etc.) to be used by a touch controller in the computing device for noise abatement. If the power adapter is not on the locally-stored table, the computing device queries a remotely-stored table (e.g., table stored in a cloud service), to obtain the touch controller data. The touch controller data is provided to the touch controller, which uses the data to configure a noise abatement strategy. An estimate of the load on the power adapter is provided periodically to the touch controller for as long as the power adapter is connected to the computing device.

FIG. 1 is an example block diagram of the system for noise abatement. A computing device 102 includes a central processing unit (CPU) 104, a memory 106, a touch input device 110 and a touch controller 112. As discussed above, charger or finger-coupled noise may occur when the power adapter 114 is connected to the computing device 102 and the user is using the touch input device 110. As discussed, the touch controller 112 may employ noise abatement techniques to minimizes and/or eliminate any effects of the noise.

The computing device 102 may include any type of computing device. For example, the computing device 102 may include a laptop, a notebook, a desktop, a tablet, a smartphone or any other type of computing device. The touch input device 110 may include a touchscreen and/or a touchpad.

The CPU 104 includes any type of processor or micro-processor, including a hardware processor, that is configured to execute instructions, which may be stored in the memory 106, to perform programmed actions. The touch controller 112 may include a hardware controller that is programmed with instructions (e.g., firmware) to control the operation of the touch input device 110. The power adapter 114 may include any type of power adapter or power supply including, for example, a USB power adapter, a micro-USB power adapter, a type-C USB power adapter or other type of power adapter.

When the power adapter (or power supply) 114 is connected to the computing device 102, the power adapter 114 identifies itself by sending identifier information such as, for example, manufacturer, part number, revision, manufacturing date, etc. The computing device 102 uses the identifier information to query a local touch parameter database 116 to look up touch controller parameters to send to the touch controller 112 to use for configuring and employing noise abatement strategies. The local touch parameter database 116 may include a table of parameters and values for each different type of power adapter. For example, the parameters may include a list of hopping frequencies, a frequency hopping strategy, filtering strategy and parameters, and other types of parameters. The parameters may be different for each different type of power adapter. The parameters may be associated with a particular adapter because, when used by the touch controller 112, they may be known to reduce and/or eliminate the noise.

In some cases, the local touch parameter database 116 may not include the parameters for the power adapter 114 or the parameters for the power adapter 114 may be out of date. In this situation or in other situations, the computing device may query a remote touch parameter database 118. The computing device 102 may include a network interface to communicate wired and/or wirelessly with remote computing devices over a wired and/or wireless network 117, including the Internet. The remote touch parameter database 118 may be stored on a remote computing device 120, including a server or other type of computing device, that may be accessed by the computing device 102 as well as multiple other computing devices. The remote touch parameter database 118 may be more regularly updated than the local touch parameter database 116 with new and/or revised parameter data related to different types of power adapters. For example, when a manufacturer releases a new power adapter, the parameters useful for noise abatement techniques may be added to the remote touch parameter database 118 and not pushed to the local touch parameter database 116. In this manner, for example, only one database needs to be updated and maintained with the most recent power adapter and corresponding parameter information, instead of having to update and maintain multiple databases distributed to multiple computing devices.

In this example, the computing device 102 queries the remote touch parameter database 118 using the identifier information from the power adapter 114. The query may return the parameters needed by the touch controller 112 to perform noise abatement techniques. If neither the local touch parameter database 116 nor the remote touch parameter database 118 contain the parameters for a particular power adapter, then the touch controller 112 may use default parameters for the noise abatement techniques.

Referring to FIG. 2, an example process 200 illustrates example operations of the computing device 102. Process 200 includes determining whether a power adapter has been connected to the computing device (202). If a power adapter has been connected to the computing device, then the computing device queries the power adapter for identifier information (204). In some instances, the power adapter may automatically provide the identifier information to the computing device without the computing device having to issue a query to the power adapter for the information.

If the adapter provides a response with the identifier information (206), then the computing device queries the local touch parameter database and/or the remote touch parameter database (208). For instance, the computing device may first query the local touch parameter database. If the information is not present in the local touch parameter database, then the computing device may query the remote touch parameter database. The query may send the adapter identifier information to the databases in order to look up the parameters corresponding to the specific power adapter.

If the adapter does not provide a response, then default parameters are used by the touch controller (207). The default parameters may be stored in the local touch parameter database or otherwise programmed as part of the touch controller.

Process 200 includes using the parameters returned from the query (210). The parameters are used by the touch controller for use with the noise abatement. If the power adapter is still connected, then the state remains the same. If the power adapter is not connected, then the computing device waits until a power adapter is again connected.

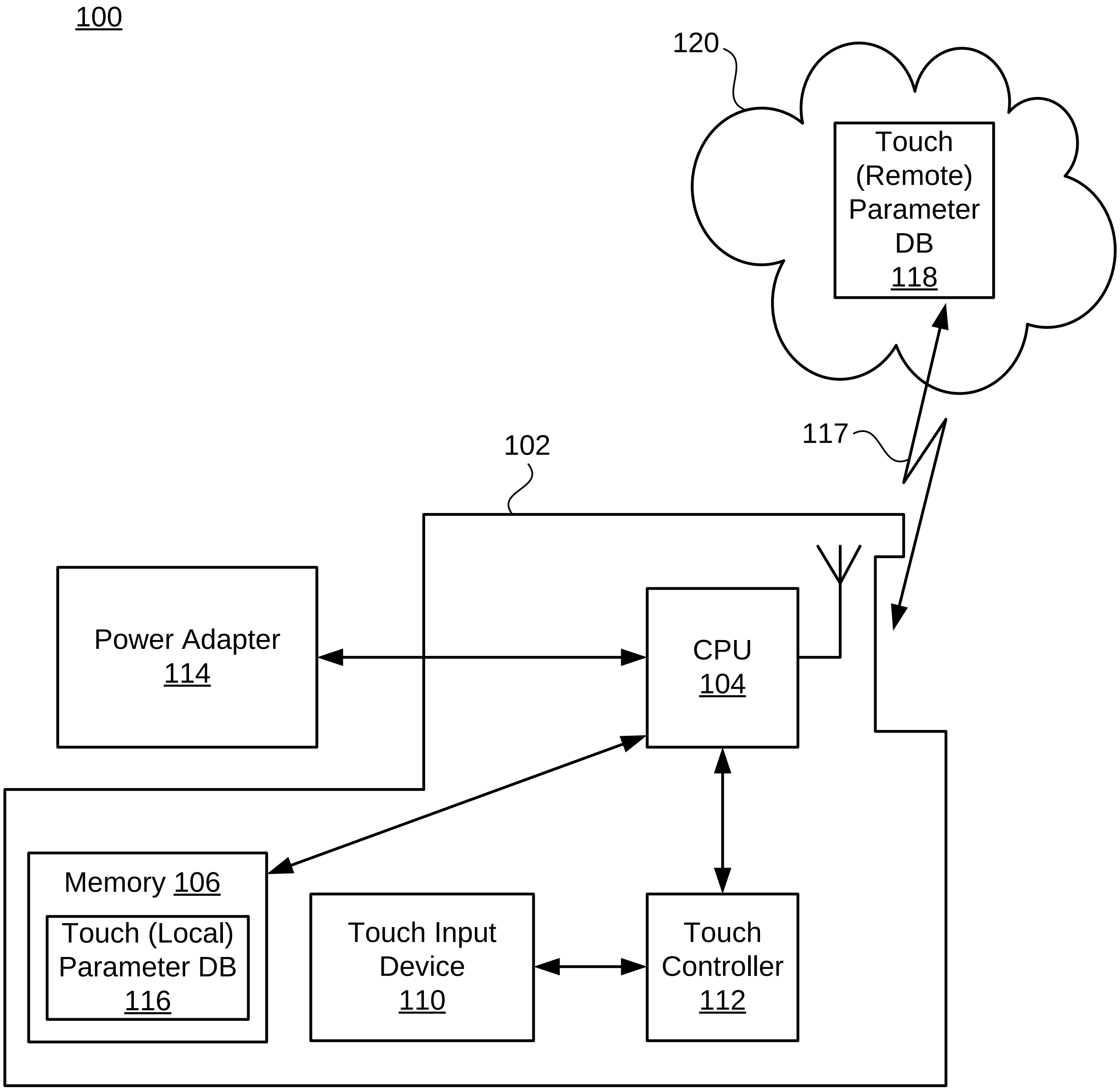
100
120
Touch (Remote) Parameter DB 118
117
102
Power Adapter 114
CPU 104
Memory 106
Touch (Local) Parameter DB 116
Touch Input Device 110
Touch Controller 112

FIG. 1

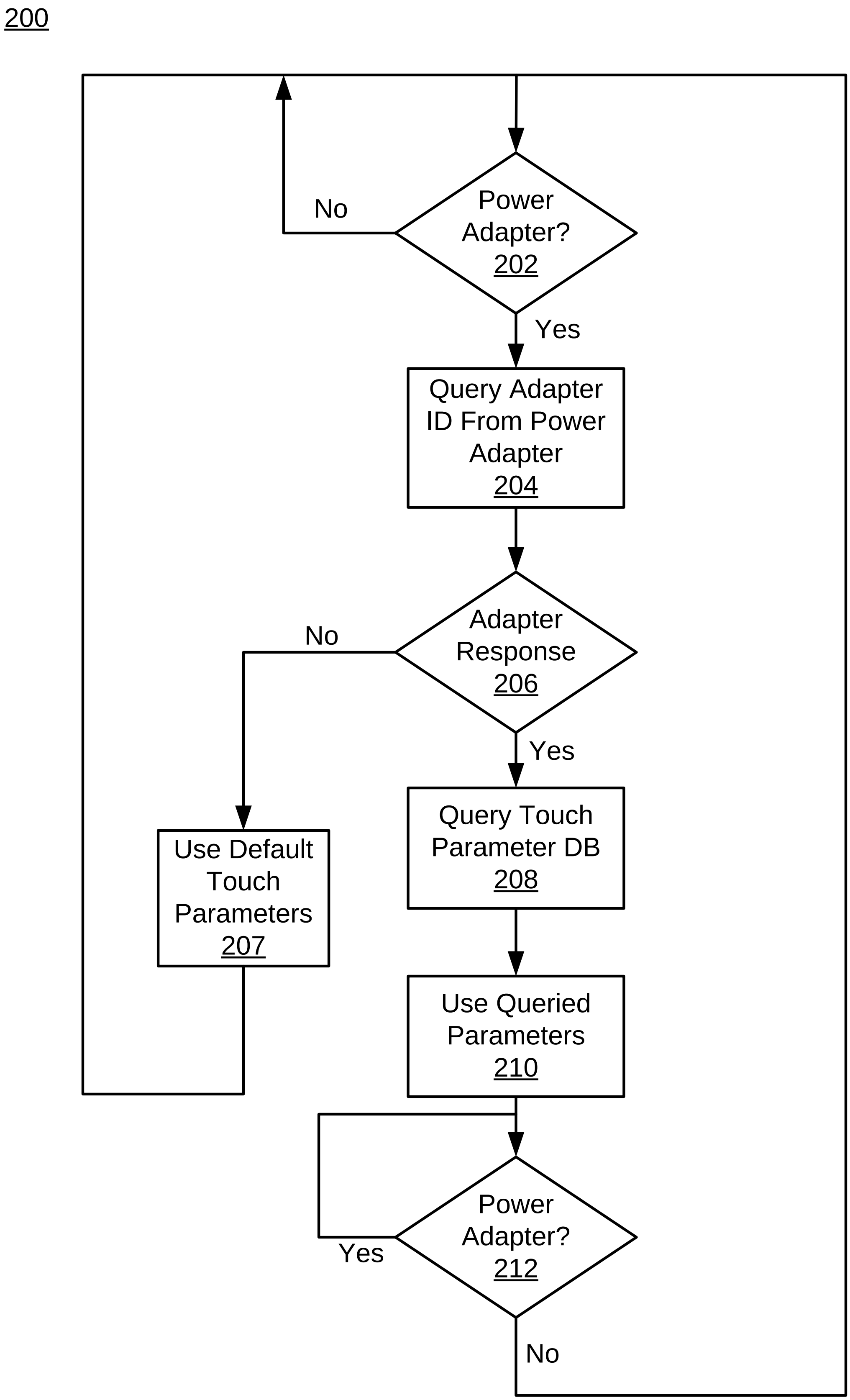
200
Power Adapter? 202
No
Yes
Query Adapter ID From Power Adapter 204
Adapter Response 206
No
Yes
Use Default Touch Parameters 207
Query Touch Parameter DB 208
Use Queried Parameters 210
Power Adapter? 212
Yes
No

FIG. 2