stitute the coordinated touch. The collective XY location of the touch is also preferably a weighted average of the constituent XY readings. The collective applied force is a total of the applied force for all the haptels involved in the coordinated touch.

[**0098**] The collective XY measurements for a coordinated touch are evaluated (step **1012**) to determine if haptels need to be added to, or removed from the coordinated touch. If the collective XY of a touch is less than a first predetermined distance of from a nearby haptel, that haptel is added to the coordinated touch, if the haptel is not already a member of another coordinated touch. If the collective XY of a touch is greater than a second predetermined distance from a haptel already a part of the coordinated touch, that haptel is removed from the coordinated touch. The second predetermined distance is preferably greater than the first predetermined distance, and the difference between these two values is preferably large enough to prevent a given haptel from being quickly added to and removed from a touch if the given haptel's distance from the touch is close to these values. The first and second predetermined distances should be large enough that the haptels being added and removed are not currently being touched. In one embodiment, a value greater than half the size of the intended touching object, such as the fingertip, is often adequate.

[**0099**] The current state of coordinated touches is periodically sent to computer **916** over serial link **914** (step **1014**). Computer **916** is preferably updated with a frequency on the order of hundreds of times per second. Computer **916** is preferably sent only the collective measurements of XY, position, velocity, acceleration and applied force. Serial link **914** is preferably checked for incoming data from computer **916** containing instructions on the haptic effects to use for each coordinated touch (step **1016**). Typically, haptic effect commands are sent after a new coordinated touch starts, and only periodically once the touch is in progress. The haptic effect might be changed subsequently depending on the state of the software executed by computer **916**. For example, after a user has started to press a virtual button, computer **916** can disable the button. The new haptic effect is preferably implemented as soon as it is received. Haptic effect command can be designated to simply assign one of many built-in haptic effects to the coordinated touch or to define a custom haptic effect (e.g., by mixing together built-in effects, transmitting a force response curve, or downloading executable machine codes which implement an effect).

[**0100**] The collective feedback force for each coordinated touch is then computed based on the collective measurements derived earlier using the haptic effect command sent from the computer (step **1018**). For a simple "button" response, for example, the feedback force can be primarily computed based on the collective position. At 0.0 mm travel the force is about 0.05 Newton, which increases linearly to 0.75 Newton over the next 0.3 mm of travel, then decreases to 0.10 Newton over the next 0.5 mm travel, and finally increases to 2.0 Newtons over the next 1.0 mm of travel, and stays at 2.0 Newtons over the remainder of travel. For a simple "disabled button" response, for example, the feedback force is 0.05 Newton at 0.0 mm of travel, increasing linearly to 2.0 Newtons over the next 0.6 mm of travel, and staying at 2.0 Newtons over the remainder of the travel. These are just two examples of haptic effects, a multitude of which are possible.

[**0101**] The feedback force for each haptel is computed by distributing the collective feedback force across all of the haptels in the coordinated touch in proportion to the amount of the applied force on each haptel (step **1020**). If the applied force of a given haptel is, for example, half of the total applied force for the collective touch, then the feedback force for that haptel will be half of the collective feedback force. Different haptels in the same coordinated touch can have different feedback force values, because they each have different applied forces, but the total of these forces will equal the feedback force for the collective touch.

[**0102**] The restoring force for each haptel is computed based on each haptel's distance from the collective position (step **1022**). Due to errors in measurements and other effects, the surfaces of the haptels in a coordinated touch can drift apart vertically. Thus a restoring force is applied which pulls each haptel in a coordinated touch back towards the collective position, proportional to its distance from that position. The restoring force is preferably greater than 2.0 Newtons per millimeter. The upper limit for this value depends on the control loop frequency and the resolution of the position measurement, among other factors. Since the haptel positions are distributed around the collective position, this restoring force does not add noticeable net force to the coordinated haptic effect. Additionally, damping can be added in proportion to the relative velocity of a haptel compared to the collective velocity. This prevents haptels from oscillating around the average position.

[**0103**] The net force for each haptel is computed, then converted to the correct actuator circuit output value and set via the digital output card (step **1024**). The net force for each haptel is simply the sum of that haptel's desired feedback force and that haptel's restoring force. The actuator force is computed by subtracting the effect of the spring at the current position and the weight of the moving assembly from the net force. The actuator force is converted to the correct output value using the calibration table recorded previously. The actuator circuits can be programmed by writing the output value to digital output card **908** by, for example, first writing out the data values, and then raising and lowering the latch bit to latch in the new data. After the actuators are set, the flow of control returns to step **1002**. The program loops indefinitely while haptic responses are being generated.

[**0104**] The operations referred to in **FIG. 10** and elsewhere herein may be modules or portions of modules (e.g., software, firmware or hardware modules). For example, although the described embodiment includes software modules and/or manually entered user commands, the various exemplary modules may be application specific hardware modules. The software modules discussed herein may include script, batch or other executable files, or combinations and/or portions of such files. The software modules may include a computer program or subroutines thereof encoded on computer-readable media.

[**0105**] Additionally, those skilled in the art will recognize that the boundaries between modules are merely illustrative and alternative embodiments may merge modules or impose an alternative decomposition of functionality of modules. For example, the modules discussed herein may be decomposed into submodules to be executed as multiple computer processes. Moreover, alternative embodiments may com-