Cognitive Motion-Dynamic Tethering of a Phased Array to an Android Smartphone

Gregory H. Huff, *Senior Member, IEEE*, Jean-François Chamberland, *Senior Member, IEEE*, and Jeffery S. Jensen, *Student Member, IEEE*

***Abstract*—An interactive framework for cognitive radio applications is proposed in which a user-equipped Android smartphone and a phased array controller are wirelessly tethered. Algorithms on the smartphone pull inertial, geolocation, and networking data to coordinate tracking and beamsteering, and the smartphone itself serves as a wireless communications bridge between the phased array's control module and a remote network server which facilitates bi-directional communication of health monitoring data, state information, and control signals. The system architecture, algorithms, and measured results from beam-steering of a 4 × 4 microstrip patch array via the Android smartphone are discussed.**

***Index Terms*—Adaptive systems, antennas, Bluetooth, cognitive radio, phased arrays, systems engineering.**

I. Introduction

SMARTPHONES AND TABLETS have arisen as a unique and ubiquitous computational resource for a wide range of purposes. Their connectivity enables applications that range from e-commerce to navigation, and in some cases they have streamlined or revolutionized common practices. Applications (*apps*) developed for this purposes can utilize the mobile device's access to the global positioning system (GPS), network information, and its own internal sensors which can monitor orientation, acceleration, and other activities. Wireless connectivity such as Wi-Fi, data services (3G, 4G, etc.) WiMax, cellular, near-field communication, and Bluetooth can also be harnessed to communicate or interact with peripheral or remote systems. This confluence of sensor data and other information can be coordinated to enhance data fusion for a common operating picture, integrated into a decision making process with human interaction, and used in a cognitively to monitor or trigger events both locally and remotely.

The cognitive capabilities of these mobile devices are used explicitly in this work to examine one of the first system-level implementations of a smartphone as a cognitive controller for a phased array system. More specifically, a concept and system framework is proposed which operate outside of the spectral-centric cognitive radio paradigm by integrating the motion-dynamic interaction with individual smartphone users into the cognitive radio framework. This takes advantage of advancements in the computational resources available in mobile devices, which were unavailable when concepts such as this were originally considered [1], but reaches beyond the basic digital design task by proposing a modularly-constructed framework for the phased array system that is based on low-cost microcontrollers and peripherally connected devices that have been wirelessly connected, or tethered, to the mobile platform [2]–[6]. The network-centric discussion is intended to illuminate a control architecture that emphasizes the antenna systems' inheritance of cognitive features from the mobile platform rather than the classic design process that is centered on the digital/analog control system.

The discussion on the cognitive smartphone controller for phased arrays follows the basic flow of control signals through the system to highlight the interconnectivity of subsystems and the role of the smartphone as a central controller. It begins by examining the system-level objectives and required (or desired) operation of the array. Each subcomponent of the phase shifting network and control subsystems is then discussed along with the communication protocols used to develop cohesiveness among algorithms in the smartphone, the network of microcontrollers, and a remote server used for health and state monitoring. A 4 × 4 phased array of microstrip patch antennas is then constructed and connected to the phase shifter network to evaluate the performance of the system. A brief review of the system concludes the work.

II. State Operation and System Overview

A. Operational Modes

Two operational modes and one benchmarking mode of the system are examined *a priori* to system and subcomponent design. It is assumed here that the smartphone has information on the physical array topology (number of elements, spacing, etc.). This initial information helps define the basic roles of subsystems and determine the communication protocols required to pass data between them. Linear progressive phasing is also assumed here, but more advanced modes of operation (sparse arrays, adaptive nulling, multiple beams, reconfiguration, etc.) are possible. These are not considered explicitly in this discussion which focuses on system design.

Manuscript received November 07, 2012; revised February 11, 2013; accepted April 18, 2013. Date of publication June 07, 2013; date of current version February 27, 2014. This material is based upon work supported by, or in part by, the US Army Research Laboratory and the US Army Research Office under agreement number W911NF-09-1-0429.

The authors are with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: prof.ghuff@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAP.2013.2267201

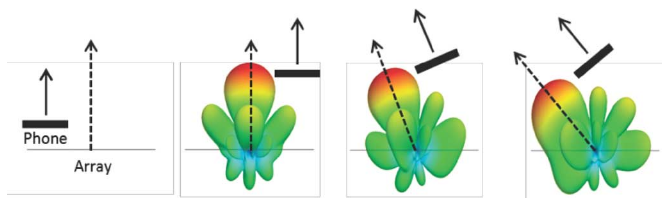


Fig. 1. Wirelessly tethering the array to the smartphone and cognitively controlling the scan angle ("*Tethered Control*" mode) using a motion-dynamic user interaction; (far-left) tethering the coordinate systems of the smartphone and phased array, (mid-left) calibrating main beam to broadside when both are coplanar, and (right) using the smartphone's orientation and motion to scan the main beam and track a point P directed normal to the smartphone.

1) Tethered Mode: This mode demonstrates cognitive control of the antenna array using the smartphone's motion-dynamic sensors and geolocation capabilities. The main beam is wirelessly tethered in this mode to the orientation and location of the smartphone, and the main beam of the phased array tracks a point P normal to the remotely located smartphone. Fig. 1 illustrates the calibration and operation of this mode, which begins in the *app* by tethering the coordinate systems of the array and smartphone using a nominal calibration which considers the coplanar alignment of the smartphone and the fixed-position antenna array. The smartphone then records location and orientation data from GPS and its inertial measurement unit (IMU), respectively. The antenna is then electronically controlled from the phone such that the main beam scans to track the smartphone's orientation with respect to the calibrated position P. This mode is defined as the "*Tethered Mode*" since the smartphone controls the main beam by cognitively scanning it to point in the user-defined direction.

2) Tracking Mode: A more generalized mode is considered for object tracking. Continuous or event-triggered changes are made remotely by the smartphone in this mode to compensate for movement or rotation of the smartphone, array, or object being tracked. The smartphone uses geolocation and orientation information again to remotely scan the main beam, but it uses its resources to cognitively track the desired target located at a moving or fixed point P. The *app* first triggers a recording of the orientation and location of the smartphone and antenna array, then the *app* starts similar to the tethered mode. It then runs as a background application that monitors the IMUs of the smartphone and array as well as the desired direction to the tracking object at P to scan the main beam accordingly. The primary difference between this and the "*Tethered Mode*" reside in determining and updating the tracked point P. An additional requirement of this mode is bi-directional communication of IMU and GPS data to monitoring the array itself; the smartphone cognitively updates this information to ensure any movement or rotation of the array is accounted for in the phasing of the array.

3) Benchmarking Mode: This is a feature of the aforementioned modes, and is a set-and-hold function to manually fix the desired scan angle of the array. This is handled remotely from the smartphone's touchscreen interface and is the primary method used for anechoic chamber measurements of the prototype system. This can be expanded through handles for control that can be passed into and out-of other *apps* or used in other processes.

B. Android Algorithm Development

The framework for a basic tracking algorithm on the Android smartphone can be synthesized from the two aforementioned modes of operation. The smartphone initializes a three-dimensional unit vector in both a relative and global coordinate system using the IMU and GPS. This vector remains normal to the plane of the smartphone in the Tethered Mode and points to a nominal point P in space located normal to the smartphone. After this calibration step, the smartphone continuously reads and updates its orientation and geolocation information to determine the required angular shift of array's main beam so it remains steered in the directed of P. These new angles are computed in a standard rotation matrix and used to determine the required element-level phase shifts. The pseudocode for tracking a point P in the tethered mode is illustrated below.

```
//Initialize
Point[]=[θ,θ, 1] // normal vector for broadside rad.
User_local=Get_orient_geolocat_data(IMU,GPS);
(Ψ,Θ,Φ,)=calc_quaternions()//set relative coord. sys.
//Track
Track_point_P=Get_orient_geolocat_point_data(IMU,
GPS, P);
(Ψ,Θ,Φ)=calc_quaternions(User_local, Track_point_P)
New_point[]=Track_point_P[]*Rot_matrix (Ψ,Θ,Φ)
(θ0,φ0)=calc_new_point_angle(New_point[]);
(βx,βy,βz)=calc_phase_shifts (θ0,φ0); //payload for
packet
//Send Update
Packet[startbyte, type, length, βx,βy,βz, checksum];
Send_to_control_curcuit_via_bluetooth(Packet[]);
Send_to_server_via_wifi(Packet[]);
```

The initialization of the system includes the creation of quaternions (yaw Ψ, pitch Θ, and roll Φ) for the smartphone's relative coordinate system using IMU and GPS data. If broadside is desired, the orientation when the array and smartphone are coplanar (similar to the example in Fig. 1). The tracking phase updates the quaternions and point P in space, and then uses this information to calculate the required location of the main beam (θ_0, ϕ_0) and progressive phase shifts $(\beta_x, \beta_y, \beta_z)$ required to point the main beam in the tracked direction (only a 2-D planar periodic array is considered here so β_z is not used). The update bundles this phase data into a variable-length packet and sends this information to the phased array controller where it is decoded for control, and simultaneously forked to the server for health and state monitoring. The packet structure discussed later in this work can facilitate a number of different physical events within the system including calibration and the remote real-time reprogramming of the controller for different array topologies or operating conditions.

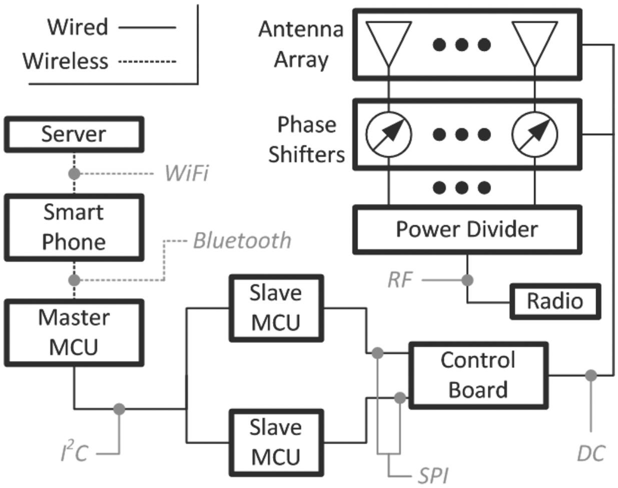


Fig. 2. Basic layout of the modules and the types of communication used to pass information and control between them.

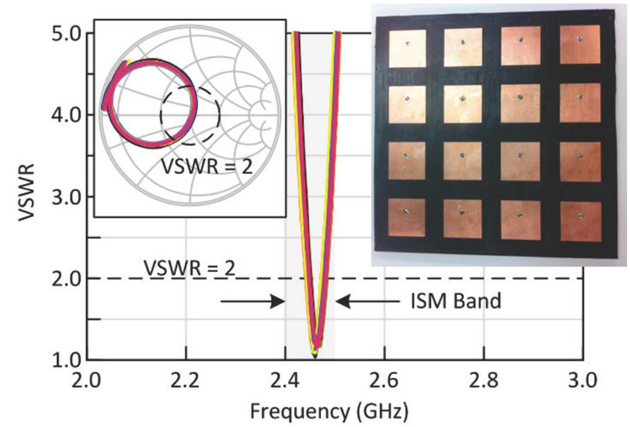


Fig. 3. Measured impedances (overlaid Smith chart) and VSWR for each of the sixteen elements of the fabricated array (overlaid photo).

C. Physical System

There are numerous possibilities available to construct the physical network of control modules which can execute the desired operation of the system. A scalable, modular, and low-cost system using off-the-shelf components is examined here to make its operation scalable and multifunctional. Fig. 2 shows the basic layout of the system in this work, and the types of communication used to connect the modules. The smartphone in this diagram shares two wireless connections; the first is the Bluetooth connection to a series of microcontrollers (MCUs) which share control information and the second is the Wi-Fi connection with a remote server to capture network diagnostic information. The "radio" block in this setup is not shown with any connection to other system components, but it can be connected into the wired or wireless networks; for this work it will only be implemented as a CW source for RF measurements.

A master-slave MCU network has wireless master nodes that reconfigure the role of slave nodes, processes control information sent from the smartphone and route packet information to slave controllers on a wired Inter Integrated Circuit ($\mathrm{I^2C}$) network. The slave nodes decode the payload to produce the DC bias voltage on a control board and other operational or sensing tasks. A bank of phase shifters feeding the antenna array through a RF power divider network receives DC voltages from the board to apply the desired phase shift at the carrier signal. In this configuration the MCU network gathers information from external sensors such as local IMUs and returns this to the smartphone. Additionally, the control board uses a Serial Peripheral Interface (SPI) to communicate with the MCUs (as opposed to $\mathrm{I^2C}$) since this can be implemented as a secondary bus for information.

III. Modules and Subsystems

A. Antenna Array

A 4×4 rectangular planar periodic array of probe-fed, square, linearly polarized microstrip patch antennas with a width W and resonant length L was designed on a 1.575 mm (62 mil) thick RT/Duroid 5880® [7] substrate with a free-space element spacing $d_x = d_y = 0.5\lambda_0$ to test the proposed control system. A design frequency within the 2.4–2.484 GHz ISM band was desired so a resonant frequency of 2.45 GHz was chosen for the patch antenna and array. The array was simulated [8] then fabricated with elements using $W = 40$ mm and $L = 39.75$ mm. The measured input impedances of the sixteen patches in the 4×4 array from 2 GHz to 3 GHz on a Smith chart and the corresponding measured VSWR are included in Fig. 3 along with a photo of the fabricated array.

The measurements are in very close agreement with the expected results (not shown) and all coupling behavior is nominally within levels associated with this element and array topology. Each element has a $\mathrm{VSWR} < 1.20$ with a center frequency $f_0 = 2.46$ GHz and an impedance variation of less than 3% at resonance amongst all patches. This is close to calculated and simulated models, and the 2:1 VSWR matched impedance bandwidth shared by all element falls within the ISM band. The slight upward shift is attributed to the mechanical milling of the array, but the matched impedance bandwidth still satisfies the design criterion so no modifications were made.

B. Linearization and Control of Phase Shifters

Commercially available electronically-controlled analog phase shifters [9] with a 0°–450° phase response over a 0–13 V DC control voltage range were chosen for the phased array system (one at the output of each port of the power divider). Each module was individually characterized using a custom LabView VI [10] which automated measurements by repeatedly cycling a DC control voltage in 100 mV increments while recording the unwrapped phase shift. A maximum variation of 3% from 9–12 Volts was observed as a maximum between the sixteen phase shifters.

This characterization provided insight into the repeatability and consistency amongst phase shifter units, but the primary goal of the activity was to develop a mapping between the applied voltage and the corresponding phase shift that could be linearized and programmed as a function in the control module. To do this, a simple fourth-order polynomial was first generated to fit the experimental data. This polynomial was then used to linearize the phase shifter control data, such that the 0–12 V range could be mapped in the control board to produce uniform

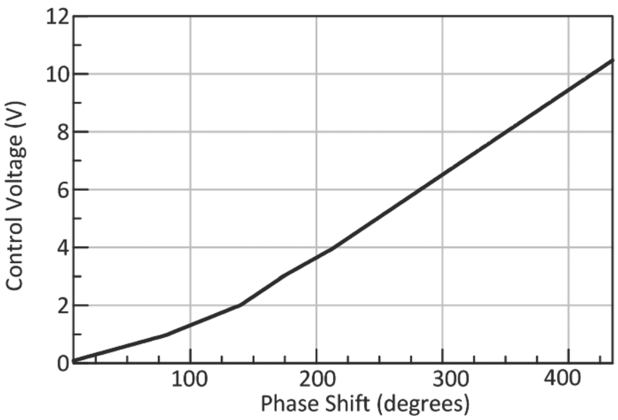


Fig. 4. Linearizing mapping function used by control board to provide uniform incremental phase shifts.

increments in the phase shift as a function of applied control signal. The mapping has been plotted in Fig. 4, and is shown as a control voltage that is plotted as a function of the desired phase shift (as the slave MCUs interprets this).

C. Control Board

The control voltage to the analog phase shifter is the first link in the control path (moving from the antenna array toward the smartphone) and can be implemented in a variety of ways from the MCU. Pulse width modulation (PWM) is a common feature of MCUs, but providing independent PWM signals for each phase shifter is explicitly avoided since the number of PWM output pins on a given MCU platform is typically quite small in comparison to other I/O. This limits the scalability and increases the susceptibility of the control signals to system noise. It is also good practice to deploy current and/or voltage buffering in stages to avoid placing a significant current draw on the limited resources of the low cost MCU, so an alternative approach as developed.

A series of digital potentiometer [11] circuits are designed to overcome the PWM limitations and provide the analog phase shifters with a high fidelity DC control signal. They are assembled using low power components and can be quickly reconfigured to meet the demands of the algorithm. They use SPI communication to write resistance values into its internal memory, operate at the TTL voltage levels from the MCU, and are capable of producing highly accurate and reproducible voltages. The digital potentiometers in this work accept commands on the SPI bus to set the internal voltage divider to a desired position (the 128-position wiper in this work), and they are capable of communicating their resistance position to a central controller for state and heath monitoring.

A series of operational amplifier circuits were designed to achieve the maximum of 12 V DC for the phase shifter since the digital potentiometers are only capable of producing a voltage range of 0 V to 5 V, so simple non-inverting operational amplifier circuits with carefully selected resistances are used to provide the appropriate amplification stage needed to both increase the range from 5 V up to 12 V and buffer the TTL output current capacity of the potentiometer. In this design, the slave node MCUs act as local-level controller which is responsible for a select subset of phase shifters. It must also be capable of reading external devices and sensors, provide data processing power, and provide open connections to other modules.

The MCU and control board are required to be modular in this work to accommodate both small and large scale antenna configurations (in subarrays). Each slave node's MCU [12] in this work is capable of handling communication for up to eight phase shifters, which is limited only by the amount of digital communications pins that the MCU has available. Two slave nodes (denoted “A” and “B”) were therefore required a sixteen element array. This could be reduced through multiplexing and other techniques, but this limits the scalability of the system and its ability to simultaneously control all sixteen phase shifters. Slave nodes don't necessarily need to communicate with each other, but in this implementation each slave MCU must receive the same information so algorithmic outcomes and control information are preserved across the array for integrity and error checking.

The master node MCU [13] facilitates communication and control. Its primary functions include communicating with the smartphone via Bluetooth, communicating with the phase shifter control modules, and maintaining inter-communication at the control level. We establish this controller as the master in a master-slave style of communication among the control modules using I^2C. The master checks on how many control modules are online and communicates operation and state and information it receives from the smartphone via Bluetooth. The master is also responsible for initially verifying data integrity before passing information on to slaves and control networks. It is also able to reconfigure the slave nodes so they can control different types of phased array topologies.

D. Smartphone

An Android [14] smartphone ([15] in this work) serves as the central controller of the phased array system in this work. It is responsible for calculating phase shifts, communicating with the server and control network, and addressing the interactive needs of the user. This has been programmed in Java [16] using the Android SDK [17] in Eclipse [18], and the resulting interface allows the user to dynamically switch between the system modes of operation while also providing interfaces to connect to the server and control network. Important real time information such as system status, phase shift, and main beam location are shown in real time to the user. Notably, all controls and information including network clock time, phase information, and system power usage are derived from the algorithms programmed into the smartphone.

E. Fully Assembled System

Fig. 5 shows the fully assembled control board, power divider/combiner, MCUs, and other components. These components are mechanically fastened onto a 0.5 in thick Lucite mounting plate using nylon hardware. This plate is affixed to a square PVC frame (0.5 in diameter), which has PVC connections below the power divider (not visible in the photo) for mounting antenna arrays, and additional connections for

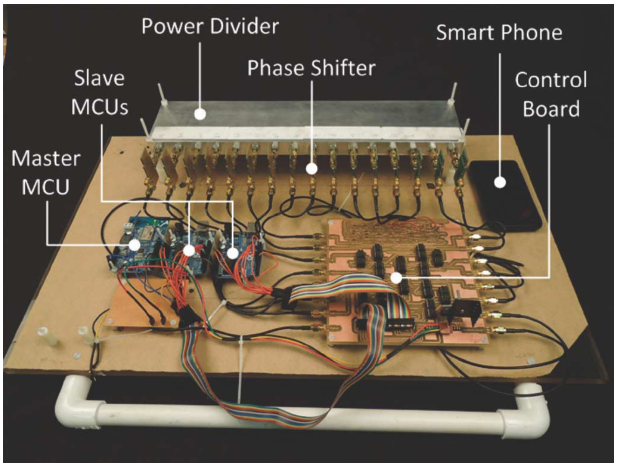


Fig. 5. Fully assembled smartphone phased array controller.

mounting and mechanical stability to the rotating pedestal in the anechoic chamber.

IV. Application (*App*) and System Operation

A. Synchronization

The smartphone serves a synchronizing module in this work that all other modules and subsystems follow. It sets the master clock and promotes an effective communications scheme between all other modules. As such, the smartphone is arguably the most important part of the entire control system so it is necessary to understand how the data will flow from one module to the other (in a layered approach) to ensure each sub-module can effectively use the same protocol. We adopt a model similar to the Open Systems Interconnection (OSI) model to construct the network architecture of the entire system. This is illustrated in Fig. 6, where both wired and wireless physical implementations are used to communicate data as it scales from physical transmission to algorithms that run at the application layer. We use existing routing and control protocols to interface the hardware with the software at the data link and routing layer.

The networking of low-cost MCUs in this design and the control architecture that relies explicitly on the smartphone as a decentralized resource for computing and synchronization are two very unique features of this work. The MCUs' roles in translating control voltages from packetized information and providing heath and state monitoring information are key processes, but the master MCU also facilitates the wireless tether to the smartphone. This deviates from traditional digital design implementations which might rely more on centralized computing resource such as field programmable gate arrays (FPGAs). The streamlined interconnection of MCUs also provides a degree of flexibility in the design since it is possible remotely reprogram or repurpose them once they are integrated into the system. This requires a detailed examination of the communication and networking protocols that link all of these different subsystems together.

B. Packetization

There exists a significant amount of communication between system subcomponents and the aforementioned modules.

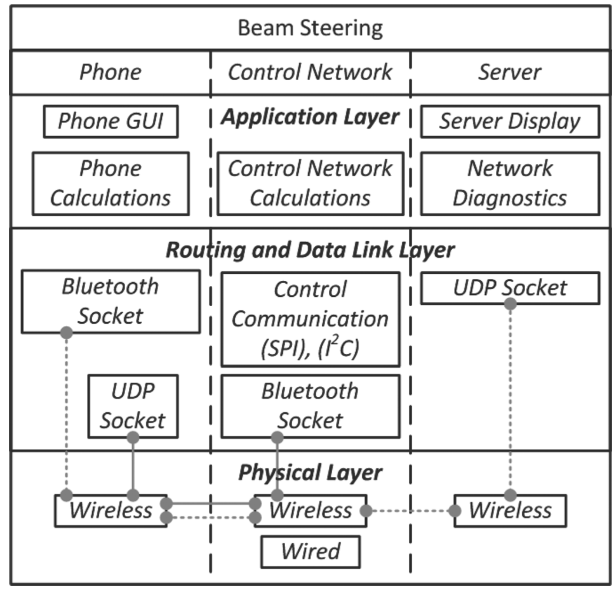


Fig. 6. Application, routing, link, and physical layers in the Android program.

Startbyte	*Length*	*Frametype*	*FrameID*	*Payload*	*Checksum*

Fig. 7. Variable-payload packet structure used for network protocol.

Custom protocols are developed as an essential part of this design so that each module may communicate in an emulating architecture similar to the OSI model. The smartphone communicates phase information to the control network and server, but also relays network information from onboard sensors to the server as well (these are not used in this discussion). Each antenna phase shifter is addressed in this model to allow individual reconfiguration of antennas (if or when desired) and accommodate various array structures and subsystems. The concept packet structure shown in Fig. 7 was used to facilitate this desired functionality.

The *Startbyte* is used to indicate to a device that a packet is starting. *Length* indicates how many bytes after the *Startbyte* exist in the packet, allowing a variable payload structure. *Frametype* is used to distinguish the type of information passed in the *Payload*. As stated, the information may be phase shifts, reconfiguration information for an antenna, power usage, or other variables. *FrameID* permits the system to use handshakes if needed emulating a *SYN/ACK TCP* approach. The *Payload*, or information, is variable in length and is essential for system operation. A *Checksum* is calculated at each reception endpoint to ensure the integrity of data as it moves through system components. If at any point the checksum is incorrect, the packet is dropped and a new packet is requested.

C. Networking and Multithreading

Socket threads are initialized to have the smartphone create a wireless connection to the server and Bluetooth master MCU. These threads are separate processes which handle data transmission and packet inspection for the aforementioned custom protocol. The system enters into a continual while loop that may

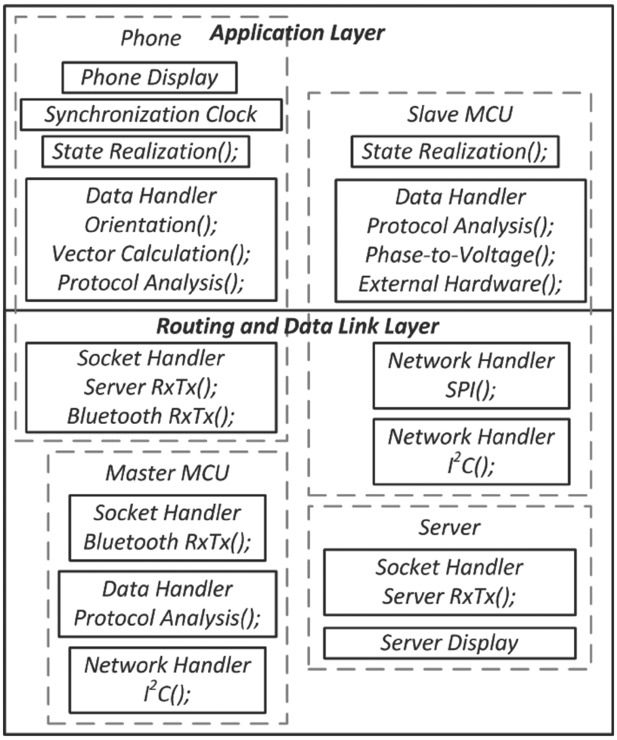


Fig. 8. High-level system operation and function representation for all modules.

be interrupted from the socket threads when new data is available; however, it continues to loop checking for user input and calculating the progressive phase shift values based on operating state. It forwards the information to the server and control network and continues to loop. At any time, the user may change the input to adjust the operating state which is processed to determine what data is calculated.

Each sub-module in this hardware arrangement is set-up to maintain specific threads in the software to help implement specific functions. Fig. 8 shows the high level system and functions for all modules. Handlers in this diagram denote threads executed by the primary operating system which act asynchronously with the overall synchronization clock and primary functions under a specific thread are noted in Fig. 8 using "*()*;". The use of the smartphone as a central controller is also illustrated in Fig. 8; it contains the synchronization clock for the entire system (i.e., the max rate in seconds at which the system can change its operating state).

The smartphone also realizes the state of operation noted by the control algorithms. Within this, the data handler realizes the state and gathers the information from separate functions to meet the state realization. It uses a socket handler to transfer the calculated information from the data handler to the appropriate connection. The master controller also contains a socket handler for the Bluetooth connection to receive packets from the smartphone; it has a data handler to process the packet information to determine data integrity. It uses a network handler and corresponding I^2C function to communicate with the slave controllers and control board.

The slave controllers are responsible for conducting protocol analysis for data integrity, converting phase shift information into DC bias voltages, and realizing any external hardware that may be connected on the I^2C bus. It also has a state realization function for other uses that are beyond the scope of this paper. A network handler is used to communicate with the control board using SPI (this communication is unidirectional). The primary SPI function accesses the respective digital potentiometer and writes the appropriate DC bias value to its memory. The server has a socket handler to communicate with the smartphone and display real time information remotely. Little data handling must be conducted as the information from the smartphone is already processed.

V. Measured Results

Both the communications platform and the control board were thoroughly tested to ensure that the system was producing the correct phase shifting behavior results prior to testing the array in the anechoic chamber. The master and slave MCUs were programmed accordingly and connected to the control board (with some additional logic) to test the protocols and functionality in hard-coded tests, then the smartphone was wirelessly tethered to the control network. Expected control voltages were observed at all sixteen ports on the control board. The discretized phase shifts from these measured voltages were then inserted into simulation [8] to test beam steering accuracy. These results indicated that the intercommunication between modules and desired control was achieved, so the "Benchmarking Mode" was chosen to control the array while testing in the anechoic chamber. The following system algorithm is programmed from the smartphone's perspective for this; it allows the user full control over the operating state of the beam steering system from the user's interface of the GUI running on the smartphone.

```
SocketThread1=new
Thread(connect_Bluetooth_socket());
SocketThread2=new Thread(connect_Server_socket());
While True:
Check_user_input();
If(!manual_user_control())
   Point[]=|object_position[]-our_position[]|
Else
   Point[] = [θ,θ, 1] //broadside radiation
If(!manual_object_tracking)
   (Ψ,Θ,Φ)=Bluetooth.read(orientation_sensor);
Else
(Ψ,Θ,Φ)=Get_orientation_data();
New_Point[]=Point[]*Rot_matrix (Ψ,Θ,Φ);
(θ0,ϕ0)=calc_new_point_angles(New_Point[]);
(βx,βy)=calc_phase_shifts( θ0,ϕ0);
Packet[startbyte, type, length, βx,βy, checksum];
SocketThread1.send(Packet[]);
SocketThread2.send(Packet[]);
```

The measurement campaign for the antenna array only examined a classical progressive phase shift for the periodic array

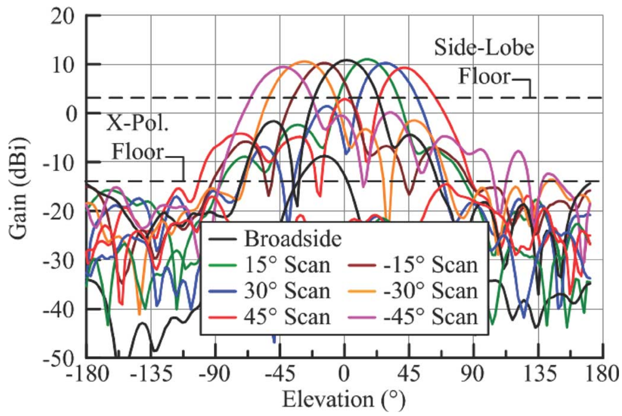


Fig. 9. Measured Co- and cross-pol E-plane radiation patterns.

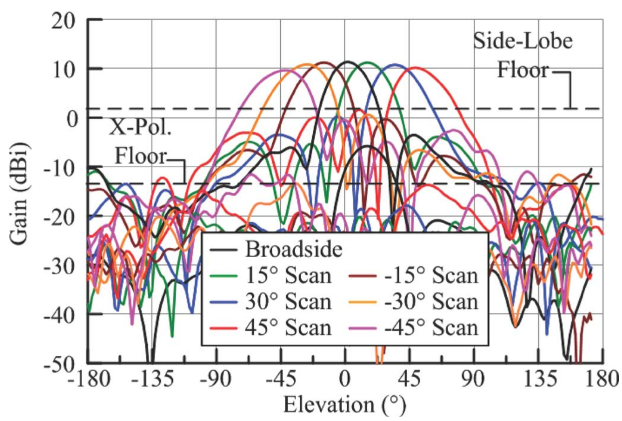


Fig. 10. Measured Co- and cross-pol H-plane radiation patterns.

based on the user-input scan angle. This provides a baseline performance metric for the system, but a range of more advanced array scanning profiles and element topologies are possible with this framework. In this work, the array was only scanned symmetrically from $\theta = \pm 45^\circ$ through broadside in both the E-plane and H-plane of the patch array using 15° increments. Figs. 9 and 10 show the measured co- and cross-pol. in the E- and H-planes, respectively. The main beam remained within 2° of the desired angle in all tests; results for the un-scanned plane were also measured and showed no anomalies so this data was not included.

A visualization tool was connected to an Apache server to remotely monitor and display real-time system information (or from a database) such as the three-dimensional beam location, phase shifts of elements, and other characteristics. Network characteristics, such as power usage, are stored in a database and can be accessed via web. The server is capable of responding to HTTP and HTTPS requests; the latter being useful for secure remote monitoring situations. The primary tool used to construct the graphical interpretation of the beam was through a tool called "Processing" [19], which like the Android environment, is written in Java. The server display in Fig. 11 includes the directional cosine angles from the calibrated axis (shown as X, Y, Z) and the progressive phase shifts are displayed as "B_X" and "B_Y" for the β_x and β_x axes respectively. In addition, a color coded map of each

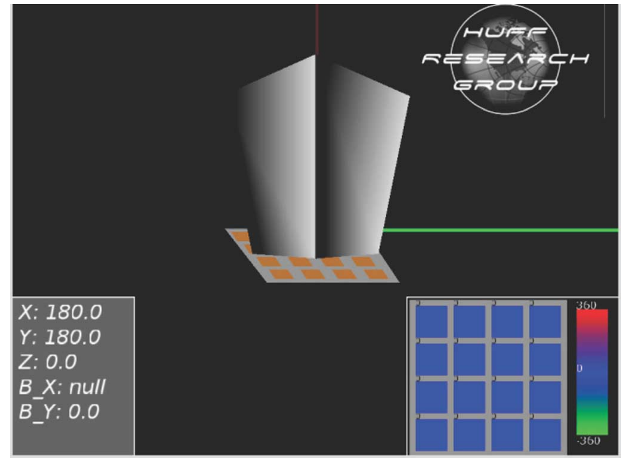


Fig. 11. Server display and real-time visualization tool created in Processing.

antenna element displays the phase shift corresponding to the current data. The large rectangular object in the middle of the screen represents the beam of the array, and moves in real-time according to movement of the main beam's direction.

VI. Discussion

There are several notable achievements in this work from a concept and operational perspective, but there are also several limitations with respect to the implementation and components used to execute the system. These are illuminated in this section in an attempt to provide a broader view of the intended contribution and highlight some potential directions for future work in this area. One of these is a practical limitation that manifests itself in the bandwidth provided by the antenna array. It is fully acknowledged that this microstrip patch antennas in this work *do not* provide continuous coverage across the entire ISM band. The intent here is to provide a detailed system-level discussion so the antenna was designed for experimental observations and chosen to remain tractable, and purely for demonstrative purposes. The discussion is application agnostic for this purpose and it is assumed that any antenna (wideband, reconfigurable, etc.) could be placed directly into this system and perform as expected.

The array topology represents another potential concern in the hierarchical view of complexity, although perhaps more from the standpoint of electromagnetic novelty [20]–[24]. The square array with equal spacing and a progressive phase shift when scanning are clearly unremarkable in the context of the complexity that these elements of the design can achieve. However, like the antenna, this was chosen purely as a well-understood and tractable implementation of an antenna array that could be used in this work without diluting the intended discussion on the system. Items such as scan resolution also arise in this context since the digital system quantizes the input to the analog phase shifter. Hence, if the PWM signal was used directly from the MCU the resolution would be limited to the n bit resolution of the PWM output ($\sim 1.4^\circ$ steps for the 8-bit PWM). Quantization is limited here only by the n bit wiper positions of the SPI digital potentiometers. These were chosen for demonstrative purposes, so this granularity ($\sim 2.8^\circ$ steps for the 7-bit digital potentiometers) can be improved by increasing the number of control bits into the phase shifter control. This is analogous to digital phase shifters, and other longstanding issues related to the quantization effects on tracking (e.g., [25]).

Lastly, it should be noted that the system design proposed in this work offers a full-scale implementation that covers the basic antenna operation, tethering motion-dynamic events to a decentralized smartphone, network concepts, and a remote server that is also tethered to the system. A discussion that simply implemented basic interaction at some rudimentary level, such as creating a simple switched event for reconfiguration or any other array purpose at an MCU from a smartphone or other wireless terminal, would simply not suffice as the focal point of a discussion on this subject. In this respect, this work intends to contribute one of the first system-level demonstrations on an emerging paradigm which is focused on the cognitive control of an antenna system beyond the spectral or data-centric viewpoints which occupy a synergistic but arguably fundamentally different design space.

VII. Conclusion

The use of a smartphone as the central control for a phased array was successfully demonstrated. By utilizing the smartphone's internal sensors and processing power, this work has demonstrated how a mobile device can provide a scalable platform and architecture for complex communications systems. To accomplish this, the smartphone's algorithms focused on a point in Cartesian space and converted this to progressive phase shifts for a 2.45 GHz phased array system. Analog phase shifters provided the necessary phase shift of the carrier signal passing to individual antenna elements. A control board and MCU network was presented as a physical bridge between the smartphone and the antenna array in order to interpret data and control the phase shifters. A physical array was designed, simulated, and tested to operate within the desired frequency band. Testing of the connected physical system illustrated that the smartphone provided directional beam steering within a 90° scan volume with less than 2% error, and real-time visualization tools were.

From an applied perspective of cognitive radio, the scalable architecture of the system provides a great deal of flexibility. The smartphone is 'cognitive' of physical changes to the array and is capable of reacting to, or compensating for, motion-dynamic activities. It is also possible to monitoring the received signal strength (RSSI), throughput, and/or bit error rate, and cognitively altering phasing or even reconfiguring antennas to alter the communication channel in a way that provides better connectivity; these topics are explicitly *not* addressed in this work which focuses on system development.

This server-smartphone connection also represents another unique perspective and offers two distinct advantages. First, the smartphone demonstrates its utility by acting as the gateway for a Supervisory Control and Data Acquisition system. Information passes directly from the system, such as power usage, and is stored on the server for trending, managing system parameters and decision making. An inherent SCADA system by an introduction of a smartphone also enables more efficient data mining. Second, the near real-time server display offers direct physical insight that is often difficult to conceptualize. Observing array parameters, such as phase shift, and physical direction of the array, electromagnetic radiation becomes a more tangible concept providing a three-dimensional spatial representation which is more easily comprehensible at a macro observation level.

Acknowledgment

The authors also wish to acknowledge material support from Hittite Microwave Corporation for their assistance in acquiring the evaluation boards for all phase shifters, Rogers Corporation for their donation of microwave laminates, and Google, Inc. for their donation of Nexus 4G smartphones. The authors also thank the reviewers for their insightful comments.

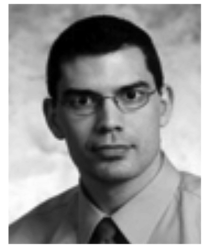
References

[1] F. J. Langley, "Commercial micro computer chips for integrated phased array control," in *IEEE Int. Microw. Symp. Dig.*, 1974, pp. 50–53.
[2] G. H. Huff, J.-F. Chamberland, S. Shakkottai, H. Waterman, J. Hanson-Daisa, A. McDonald, and N. M. Morales, "An Android controlled phased array," presented at the IEEE Int. Symp. Antennas Propag. USNC/URSI Nat. Radio Sci. Meeting, Spokane, WA, USA, Jul. 2011.
[3] J. Jensen, J.-F. Chamberland, and G. H. Huff, "Smart phone enabled cognitive control of a phased array," presented at the USNC/URSI Nat. Radio Sci. Meeting, Boulder, CO, USA, Jan. 2012.
[4] J. S. Jensen, J.-F. Chamberland, and G. H. Huff, "Development of a smart phone enabled cognitive controlled phased array," presented at the IEEE Int. Symp. on Antennas and Propag. and USNC/URSI Nat. Radio Sci. Meeting, Chicago, IL, USA, Jul. 2011.
[5] J. S. Jensen, "A Cognitive Phased Array Using Smart Phone Control," M.S. thesis, Dept. Elect. Comp. Eng., Texas A&M Univ., College Station, TX, USA, 2012.
[6] J. Jensen, J.-F. Chamberland, and G. H. Huff, "Beamsteering, tracking, phasing using an android smart phone," in *Proc. Antenna Applicat. Symp.*, Monticello, IL, USA, Sep. 2012, pp. 112–124.
[7] Rogers Corp., RT/duroid 5880 [Online]. Available: http://www.rogerscorp.com
[8] Ansoft, HFSS© v14.0. Pittsburgh, PA, USA, 15219.
[9] Hittite Microw. Corp., HMC928LP5E 450° Analog Phase Shifter SMT [Online]. Available: http://www.hittite.com
[10] National Instruments Corp., LabView. Austin, TX, USA.
[11] Microchip Tech. Inc., MCP4241 7/8-Bit Single/Dual SPI Digital POT With Non-Volatile Memory [Online]. Available: http://www.microchip.com
[12] Arduino, Uno [Online]. Available: http://www.arduino.cc
[13] Arduino, BT [Online]. Available: http://www.arduino.cc
[14] Android, [Online]. Available: http://www.android.com/
[15] Google, Inc., Nexus [Online]. Available: http://www.google.com/nexus/
[16] Oracle Corp., Java JDK v7. Redwood Shores, CA, USA.
[17] Google, Inc., Android. Mountain View, CA, USA.
[18] The Eclipse Foundation [Online]. Available: http://www.eclipse.org/
[19] Eclipse Processing [Online]. Available: http://www.eclipse.org/
[20] B. A. Munk, *Finite Antenna Arrays and FSS*. Hoboken, NJ, USA: Wiley-IEEE Press, 2003.
[21] T. A. Milligan, *Modern Antenna Design*. Hoboken, NJ, USA: Wiley, 2005.
[22] , S. Chandran, Ed., *Adaptive Antenna Arrays: Trends and Applications*. New York, NY, USA: Springer, 2004.
[23] B. Allen and M. Ghavami, *Adaptive Array Systems: Fundamentals and Applications*. Hoboken, NJ, USA: Wiley, 2004.
[24] R. C. Hansen, *Phased Array Antennas*. Hoboken, NJ, USA: Wiley, 2009.
[25] D. E. N. Davies and I. D. Longstaff, "Some new studies of angular resolution for linear arrays," *Radio Electron. Eng.*, vol. 32, no. 6, pp. 341–350, Dec. 1966.

Gregory H. Huff (S'03–M'07–SM'11) was born in Oklahoma City, OK, USA, in 1975 and received the B.S., M.S., and Ph.D. degrees in Electrical Engineering from the University of Illinois at Urbana-Champaign, IL, USA, in 2000, 2003, and 2006, respectively.

He has been with the Electromagnetics and Microwave Laboratory in the Department of Electrical and Computer Engineering at Texas A&M University in College Station, TX, USA, since 2006 and is currently at the rank of Associate Professor. His transdisciplinary research in multifunctional antennas and superconfigurable systems blends concepts from material science engineering, aerospace engineering, and other focus areas to examine biologically inspired concepts for reconfigurable antennas and smart skins, tunable fluidic and nanoparticle material systems, and collaborative beamforming in autonomous and unpiloted vehicles, and other enabling technologies including the use of smartphones and tablets in embedded systems for adaptive wireless sensor networks.

Prof. Huff was a recipient of the Presidential Early Career Award for Scientists and Engineers (PECASE) award through the Department of Defense in 2008 and the NSF CAREER award in that same year. Prof. Huff has been awarded best paper and presentation awards as an author and co-author, including the IEEE AP-S H. A. Wheeler Applications Prize Paper Award in 2004, and teaching awards including the IEEE AP-S Donald G. Dudley, Jr. Undergraduate Teaching Award in 2010. In addition to numerous other professional and scholarly activities, Prof. Huff serves as the faculty mentor for the IEEE student chapter at Texas A&M University and is involved in many project-based initiatives geared towards increasing undergraduate students in multidisciplinary



Jean-François Chamberland (S'98–M'04–SM'09) received the Ph.D. degree in 2004 from the University of Illinois at Urbana-Champaign, IL, USA, the M.S. degree in 2000 from Cornell University, Ithaca, NY, USA, and the B.Eng. degree in 1998 from McGill University, Montreal, Canada, all in electrical engineering.

He joined Texas A&M University in 2004, where he is currently an Associate Professor in the Department of Electrical and Computer Engineering. His research interests include communication systems, queueing theory, detection and estimation, and statistical signal processing.

Dr. Chamberland was the recipient of a Young Author Best Paper Award from the IEEE Signal Processing Society in 2006. He also received a Faculty Early Career Development (CAREER) Award from the National Science Foundation in 2008.

Jeffery S. Jensen (S'14) received the B.S. and M.S. degrees in electrical engineering in 2010 and 2012, respectively, from Texas A&M University, College Station, TX, USA, where he is currently working toward the Ph.D. degree in the Electromagnetics and Microwave Laboratory, Department of Electrical and Computer Engineering.