Reviewer #1, Concern #1: ***On pg. 3 (2) shows the multistatic beat signal -> Equation (2) shows ... On pg. 4, briefly describe the calibration process in [24] and discuss the sensitivity of the proposed approach to different environments/users in terms of calibration.***

Author Response: The authors are grateful for the comments and suggestions of reviewer #1. The manuscript has undergone major revisions to update the format, correct mistakes, and include a more rigorous explanation of details. The error pointed out by the reviewer has been corrected in the updated manuscript. Additionally, the calibration process is now described in greater detail along with a discussion on the versatility and sensitivity of the proposed approach.

Reviewer #1, Concern #2: ***In section IVB, to reach audience interested in HCI aspects, use more general terms such as position on the x, y axis, velocity, etc. These more general terms can then be linked to the more specific Radar way they are calculated such as Doppler velocity. The flexibility of mapping different parameters is interesting, but a concrete example would help give a clearer idea of how the sensor could function as a musical instrument.***

Author Response: The authors agree with the reviewer's suggestion. Section III (formerly Section IV) has been updated to use more accessible terms to the general HCI community and more explicitly establish the connection between the radar-specific terms and their physical (x, y, velocity) meaning. Furthermore, we detail the thought process behind the selection of our control parameters and how they can be mapped to music. Specifically, we provide a concrete example of how the features are mapped to musical parameters. To further clear up misunderstanding and confusion, the premise of this article is explicitly addressed as a novel set of tracking algorithms, utilizing the example of a musical interface as one application out of many that are enabled exclusively by our proposed algorithms. The rewritten manuscript reflects this shift in premise and attempts to articulate our approach and novelty more clearly.

Reviewer #1, Concern #3: ***On pg. 6 "high-hand velocities", be more precise on what type of musical gestures are supported. For example, are percussive gestures requiring higher velocities supports or more continuous Theremin-like gestures. The smoothing performed by the enhanced gesture tracking might also cause problems when more short, percussive gestures might be desired. More discussion would be helpful.***

Author Response: We appreciate this remark from the reviewer. The reviewer is right; the technique by which our musical interface is played is largely unaddressed in the previous manuscript. Our updated manuscript includes a detailed discussion of the control mechanisms of our interface in Sections III-A and IV-B. We have also addressed the type of gestural control allowed by our interface and its limitations in Section VI. The relevant portions of the updated manuscript are included below:

From Section III-A:

> "In this section, we introduce the simple approach to spatiotemporal sensing for contactless musical instrument interface. While our system generally tracks the 2-D position and velocity of the user's hand, we have identified three underlying features to achieve fine control of the musical interface: range, cross-range oscillation, and velocity. By the geometry given in Fig. 1, we define

the range as the position of the hand along the z-axis, i.e. the vertical displacement between the sensor and the user's hand. Similarly, cross-range is defined as the position of the hand along the y-axis. Subsequently, cross-range oscillation is the rate at which the hand oscillates in the cross-range direction. Velocity is given by the velocity of the hand with respect to the range z-axis. These parameters are selected such that the output musical interface will be controlled primarily by the range of the musician's hand and secondarily by the cross-range oscillation and velocity. However, these parameters can be assigned by the user based on preference using the MIDI interface. Throughout the remainder of this article, we will refer to these parameters as features extracted from the radar beat signal."

From Section IV-B:

". . . the primary mechanism to control the output of the proposed musical interface is the range (z-position) of the user's hand. Using the built-in audio output tool and the MIDI output, the range of the user's hand controls the note selection directly. Unlike the Theremin, which allows for continuous note selection, our interface quantizes the user input into predefined subregions corresponding to notes defined by the user. The subregions and allowed notes can be programmed by the user in the interactive MATLAB GUI. To play the desired note, the user must move their hand vertically to the position corresponding to that note. Similarly, the secondary parameters, cross-range oscillation, and velocity can be adjusted by the user by oscillating their hand back-and-forth in the y-direction or moving to the next note with a high or low velocity. The built-in audio output tool employs the cross-range oscillation to control a vibrato effect (low-frequency modulation of the audio signal). Thus, using this tool, the user can select the desired note by varying the range and perform vibrato at a desired rate by oscillating their hand at the same rate. Alternatively, the MIDI output tool provides the cross-range oscillation and velocity as MIDI parameters to be specified by the user in a virtual instrument environment connected to the MIDI output of our musical interface. Hence, our proposed algorithms are implemented to operate similarly to a MIDI keyboard with the hand range controlling the note selection and cross-range oscillation and velocity acting as MIDI parameters for the user to assign."

From Section VI:

The most direct musical interface comparison to our framework, is the Theremin, as both are controlled by the hand's proximity to the sensor. The pitch of the Theremin is controlled continuously by the hand's vertical location, whereas our interface tracks the range of the hand digitally and selects a note from the user-defined scale. While the Theremin uses two antennas, one for volume control and the other for pitch control, a total of two degrees of freedom, our framework offers three degrees of freedom (range, cross-range, and velocity), thus providing three controllable parameters. As previously mentioned, the musical interface promoted in this article supports Theremin-like gestures for note selection and parameter control. However, high-velocity percussive gestures could be implemented using our high-fidelity tracking algorithms, with some limitations. Small values of the weighting vector, **a**, in the particle filter algorithm can result in an excessively smoothed and overly damped system limiting the ability of the system to track sudden movements. Depending on the desired application, finely tuning this parameter is essential for enabling proper gestural control. Our proposed interface is an evolved Theremin,

utilizing a modern mmWave sensor for precise tracking in 2-D space (expansion to 3-D can be easily implemented with the proper hardware)."

Reviewer #1, Concern #4: ***The modified particle filtering model is unusual in that it bypasses the need for a motion model that is typical in tracking applications. Was a more traditional motion-based approach tried? Some discussion on this would be a nice addition.***

Author Response: The reviewer makes points out a necessary clarification lacking in the original manuscript. We have updated the manuscript by including a brief discussion on the selection of the particle filter as opposed to more traditional techniques relying on a motion model. In our experimentation, we have employed different tracking algorithms with varying success. Other traditional approaches yielded either sporadic localization or overly damped, sluggish estimation, especially when the movement is nonlinear and velocity changes often. We believe that these traditional methods suffer under frequent change of direction because of a certain element of nonlinearity introduced into the motion over time compared with the assumed motion model. The particle filter overcomes this downfall and allows us to incorporate the Doppler-corroboration into the iterative process. The relevant portion of the updated manuscript is included below:

From Section III-C:

"While traditional methods such as the extended Kalman filter (EKF) employ a motion model, our implementation of the particle filter bypasses the need for a deterministic motion model. The particle filter is selected for this application as other traditional approaches have demonstrated poor tracking performance in our experimentation, yielding either sporadic localization or overly damped, sluggish estimation. Additionally, the particle filter is advantageous as it can track non-linear dynamics and does not require prior knowledge of the motion model or noise parameters for robust localization."

Reviewer #1, Concern #5: ***In the training approach to image enhancement, would the model have to be retrained when the space is changed?***

Author Response: The authors are grateful for this question posed by the reviewer as it reveals that additional explanation would be helpful regarding the application of the image enhancement FCNN. The updated manuscript provides a more detailed discussion on this phenomenon as well as our experimental results and intuition. In general, the enhancement FCNN is trained on a specific "space" or region of interest (ROI) wherein the hand is always located. As the reviewer alludes to in their question, if the user changes the ROI, the FCNN may not perform well. From our experimentation with the 77 GHz automotive radar employed in this article, limiting the ROI to [-0.15 m, 0.15 m] and [0 m, 0.5 m] in the y and z directions, respectively, provides a robust solution our results consistently show that the hand does not provide adequate reflection to the device outside of this space, due to low radar cross-section (reflectivity) of the hand compared to noise and limited beamwidth of the radar itself. If the user changes the ROI to include additional space in the y or z directions, the reflections will not be high enough to yield detection and the FCNN will not even be used outside of the predefined ROI. On the other hand, if the user defines the ROI as a smaller space, the performance of the FCNN may degrade. From our tests, however, with several smaller ROIs, the FCNN still provides similar performance. However, given a different device with

more antennas, etc., the FCNN generally would need to be retrained. Our contribution is the explanation and intuition behind the training process as well as a full software implementation for other research teams. The relevant portion of the updated manuscript is included below:

From Section VI:

> "Since the FCNN is only trained on images within the expected region, extending the ROI outside of the trained region results in performance degradation. If the ROI is changed, the FCNN should be retrained accordingly. In contrast, the simple methods are highly flexible but cannot compete with the performance of the enhancement techniques. However, we have studied the limitations of the particular TI mmWave radar device and found that if the hand is placed outside the ROI defined in the previous section, it will not be detected. Due to device SNR and beamwidth, for most hand sizes, the reflections back to the radar will not be strong enough for detection. Additionally, we have tested the proposed FCNN in smaller ROIs and found similar results without retraining. For other array topologies, the proposed methods can be easily applied, although the FCNN will need to be trained accordingly."

Reviewer #1, Concern #6: ***The authors mention in Section VI that the response time averaging 29 ms for gesture movement to MIDI output is comparable to many MIDI interfaces. To the best of my knowledge, this is inaccurate. Most MIDI interfaces have very minimal latency (1-2 ms) in terms of sending MIDI signals. If the sound generation is performed on a computer, this typically adds 10-15 ms of latency, but that is more of an issue of software and the synthesis engine used NOT the control MIDI signal. 20 ms is considered the minimum needed for gesture control, so the proposed sensor especially when the enhancements are applied is a bit on the slow side when considering the additional latencies that are incurred typically after the control signal is received. When considering latency, there is a component having to do with efficient computational implementation as well as an inherent latency based on the computation performed. For example, using large windows when computing the Short-Time-Fourier Transform introduces a latency that is a function of the window size that no amount of fast computation can remove. It would be good to be more specific about what parts of the processing introduce inherent latency vs which ones are just a matter of specific implementation in MATLAB vs let's say a more efficient embedded device.***

Author Response: The authors appreciate the input from the reviewer and guidance in properly measuring the device latency. In the previous manuscript, the latency metric was measured incorrectly, including computer and synthesis engine latency. Removing these factors reduced the measured latency by around 15-17 ms. Additionally, the software platform has been further optimized and now employs a more efficient GPU-based implementation that further reduces the computational complexity while retraining the simplicity of the MATLAB implementation for the benefit of the reader. Furthermore, changes have been made throughout the manuscript to promote the novelty of this article as an algorithmic advancement rather than a new competitive product to the existing market. However, as we have demonstrated latencies of under 5 ms for all the methods, we believe the computational complexity is competitive compared with existing MIDI controllers. Additionally, we have included a more thorough discussion of the latency and how it could potentially be improved using an embedded solution. The relevant portion of the updated manuscript is included below:

From Section VI:

> "The average latency of each method, $\bar{\tau}$, is measured as the time duration between the new sample being captured and the estimation process being completed on that sample. The resulting estimates are streamed across the MIDI port or sent to the built-in audio signal generation tool. Additional latency contributed by the subsequent synthesis engine is highly dependent on the software used and device under test; thus, it is not considered as part of the latency due to our methods."

> ". . . While the software package presented in this article is meant to serve as a framework for demonstrating and prototyping the proposed tracking and super-resolution algorithms, the inherent latency of the signal processing steps is a key issue in HCI and must be addressed. In our research, the largest contributor of latency in our proposed system is the hand-off between the radar device and MATLAB over UDP and shared memory, at an average of 1.93 ms. Rather than streaming to data MATLAB, a real-time solution can be implemented on the TI radar device's built-in DSP, thus providing a more efficient throughput as the DSP has direct access to the samples as they are taken. Additionally, several steps in the signal processing chain will increase in efficiency with an embedded solution. Employing small window sizes, $N_z = 16$ and the number of FFT spatial points is 64, the DPF and FFT computation times can be further reduced compared to the relatively inefficient MATLAB implementation. We would also like to note that a significant decrease in latency was achieved by optimizing the implementation using GPU accelerated coding. A similar approach could be taken on an embedded solution leveraging the highly parallelizable nature of many of the steps in the signal processing chain (FFT, CNN, Gaussian distribution computation). Comparing the computational efficiency among the algorithms, the latency cost for the more robust algorithms is insignificant in proportion to the performance gain, even in the MATLAB implementation. In latency tests, the average response time using the FCNN-DPF was 3.96 ms from user input to MIDI signaling. While most MIDI interfaces outperform this metric, we believe our framework demonstrates a competitive throughput cycle time compared to existing technology and can be further improved by a more efficient implementation."

Reviewer #2, Concern #1: ***The concept is based on gesture radar that was originally proposed in Google's Project Soli and has since then extended by various other research groups. The musical application has also been demonstrated by Soli [16]. So, the novelty in this paper is an improvement toward 2-D localization.***

While I see some merit in the manuscript, it offers little advancement and novelty in terms of (radar) signal processing. For example, the use of deep learning for gesture radars is already there in Project Soli. Further, Yimin Liu's work "Micro hand gesture recognition system using ultrasonic active sensing" also makes use of learning methods for gesture recognition with ultrasonic sensors. In Sevgi Z. Gurbuz's recent work on sign language gesture recognition, fractal complexity analysis has been employed. Finally, Anibal T. De Almeida's recent work on foot gesture recognition already employs machine learning algorithms. Compared to these, the use of FCNN in the manuscript is hardly novel.

Author Response: The authors are grateful for the contribution and comments of reviewer #2. The reviewer correctly points out that the novelty promoted in our manuscript is intended to be the improvement in 2-D localization of previous techniques by using exclusively a mmWave radar sensor. However, we additionally would like to emphasize the novelty of our deep learning FCNN algorithm for spatial super-resolution on radar images. Additionally, it is important to establish the difference between gesture classification: determining the "type" or class of the gesture based on its input over time, and localization: determining and robustly tracking the position of a target across time. The papers listed by the reviewer offer relevant novelties in the arena of gesture classification using machine learning algorithms. This is the most common application of machine learning application, especially in radar. For most of these techniques, a CNN or SVM is employed in an attempt to learn the unique differences between the reflected signals for each class of gesture. Generally, a classification algorithm receives an image as the input and outputs the detected class. In contrast, our approach does not attempt to classify the gesture of the musician from a predefined set of gestures. Rather, we propose a novel image-to-image regression approach that attempts to "enhance" the image and improve the image quality and resolution. While the literature referenced by the reviewer consistently employs machine learning algorithms, our FCNN performs an entirely different task and requires a much different training procedure. Compared with existing works in radar signal processing, our proposed machine learning algorithm is the first to implement an FCNN for increased tracking resolution via spatial super-resolution. The manuscript has been updated to clarify the difference and better present our novelty compared to the existing methods using both CNNs for classification and FCNNs for image enhancement. The relevant portion of the updated manuscript is included below:

From Section I:

> "It is important to note our approach is contrary to gesture classification, wherein the objective is to determine the class of a sample from a set of predefined classes as in [14], [15]; rather, we apply a novel fully convolutional neural network (FCNN) to preserve the geometry of the image and perform super-resolution for improved localization. Prior work on resolution improvement using FCNNs has been limited to the far-field domain with large apertures [18]; however, our novel approach unifies FCNN-based super-resolution with near-field imaging on a small (8-channel) array and is shown to improve hand-tracking performance significantly."

Reviewer #2, Concern #2: ***The other signal processing aspects also lack any non-trivial extensions. The paper uses MIMO-FMCW which is the same concept employed in automotive radars. It is unclear which domain the orthogonality has been,aintained. Range migration methods and particle filters are very standard methods that have been adapted for the problem without any non-trivial modifications.***

Author Response: We grateful for the reviewer's remark and agree that the paper needs to clearly articulate the novelty our work promotes and how it fits into the existing literature. The contributions of our work towards radar signal processing are the super-resolution techniques and modification of the particle filter to fully leverage the richness of the radar beat signal. We have updated the manuscript to reflect the focus of our contributions on the improvement attained by the super-resolution FCNN. Additionally, the reviewer's comment on which domain the orthogonality of the MIMO array is leveraged has been addressed with a discussion on the time-division multiplexing (TDM) MIMO mode.

Reviewer #2, Concern #3: ***The signal models in Section III lack rigor, E.g. there is no distinction between discrete- and continuous-time models. Sampling rates remain unclear. Which signal is a vector, what is the radar target model, etc. – all needs to be specified rigorously.***

Author Response: The authors agree with the reviewer that our discussion of the FMCW-MIMO radar signal model must be more rigorous. We have addressed this concern by rewriting much of this section and clarifying the differences between the spatial and time domains as well as discussing the various sampling criteria, spatial resolution, and feature extraction at length.

Reviewer #2, Concern #4: ***Algorithm 1: Specify what are the inputs and outputs. Write each step mathematically.***

Author Response: The authors are grateful for the comment. Algorithm 1 was not written rigorously in the previous version of the manuscript and has been updated to include each step in mathematical form and specify the inputs and outputs of the algorithm. The relevant portion of the updated manuscript is included below:

Algorithm 1: Modified Particle Filter Algorithm

input : $\boldsymbol{r} = [\hat{y}, \hat{z}]^T$

output: $\boldsymbol{s}_n = [\tilde{y}, \tilde{z}]^T$

1 $\boldsymbol{X}_n \leftarrow$ rows of $\boldsymbol{X}_{n-1}$ sampled using weights $\boldsymbol{w}_{n-1}$;

2 $\boldsymbol{X}_n \leftarrow \boldsymbol{X}_n + \boldsymbol{A}(\boldsymbol{r} - \boldsymbol{s}_{n-1}) + \boldsymbol{\psi}$;

3 $\boldsymbol{w}_n \leftarrow e^{-\frac{1}{2}(\boldsymbol{X}_n - \boldsymbol{s}_{n-1})^T \boldsymbol{\Sigma}_w^{-1} (\boldsymbol{X}_n - \boldsymbol{s}_{n-1})}$;

4 $\boldsymbol{s}_n \leftarrow \frac{1}{\mathbf{1}_N^T \boldsymbol{w}_n} \boldsymbol{X}_n^T \boldsymbol{w}_n$;

Reviewer #2, Concern #5: ***Minor comments: The writing and organization require improvement:***

- ***Do not start sections directly with subsections. Introduce the section topic first.***
- ***Put space before units: Use 440 Hz instead of 440Hz.***
- ***Introduction has a paragraph that is almost 1.5 columns. This kind of organization should be avoided.***
- ***References have not been uniformly or properly formatted. Please strictly follow the same IEEE style throughout.***
- ***Add a notation para at the end of the Intro.***

Author Response: We would like to thank the reviewer for the contributions and suggestions to improve the quality and readability of the manuscript. The manuscript has been majorly revised and rewritten to accommodate the changes suggested by the reviewer.

1) Given the length limitations, we have removed the section titled "Music Theory for Engineers" and included some of the relevant topics previously in this section in the introduction. Hence, we have included a thorough discussion on several key issues and improved the formatting substantially.
2) We have addressed the previous issues pertaining to sections being started with a subsection by including paragraphs to introduce the section topic first.
3) The units throughout the paper have been properly formatted.
4) We have modified the introduction to avoid the excessively lengthy paragraph and split the discussion into several tractable paragraphs.
5) The major issues in the references have been identified and addressed to strictly adhere to IEEE styling.
6) We have included a notation paragraph at the end of the introduction section detailing the notation employed throughout the paper.