AUTHORS RESPONSE TO REVIEWER COMMENTS

Comments from reviewers have been presented in **bold text**; and comments from reviewers #1 and #2 have been listed as C1.X and C2.X, respectively, with ascending numbers used to label these comments. Responses to comments are in plain text and have been labelled using a similar format (e.g. R1.1 indicates the response to comment 1.1). Where revisions have been made they are referred to using line numbers of the revised manuscript, and these changes have been highlighted in the revised manuscript using **bold red text**.

Reviewer #1

C1.1: Fox presents a comparison of 4 different algorithms that are used to solve for the kinematics and ground force trajectories of a model follow experimental measurement and yet are dynamically consistent. Specifically the analysis examines the solution times, residual forces and moments, joint kinematics, and joint kinetics that result from applying RRA, RRA3, AddBiomechanics, and Moco Track to experimental measurements of Hamner et al.'s 5 m/s treadmill running data. The analysis shows that there are meaninful differences between these different methods across each of the categories analysed.

In addition to the paper Fox has made his scripts publicly available and performed his analysis on publicly available data. This extra contribution is of great value and will serve as an excellent basis to fairly evaluate improvements made to both the settings of the algorithms he has tested, and to evaluate the performance of new algorithms. There has been a need for a paper exactly like the one Fox has written, and I'm glad to see that he has written it.

R1.1: Thank you for this and your other comments on the paper. I believe that these (alongside reviewer #2's comments) have improved the clarity of the paper, discussion of the results, and usefulness of the associated analysis code.

C1.2: Expectations: It is easy for me to ask for something that takes an unreasonable amount of effort to address. It is not my intention to force you to do an unreasonable amount of work. I expect that you give each of my comments consideration and respond thoughtfully. If it's really not possible to meet some of my requests in the time you have, please say so.

R1.2: Throughout my responses I have tried to implement all suggestions. Where any aspect has not been fully addressed, I have attempted to provide some discussion on the reasoning behind this.

Paper

C1.3: It would be helpful to have a couple of paragraphs that place the OpenSim tools to improve dynamic consistency in the context of the wider literature. With some

searching I've found some relevant literature that could be mentioned:

André Venne, François Bailly, Eve Charbonneau, Jennifer Dowling-Medley & Mickaël Begon (2023) Optimal estimation of complex aerial movements using dynamic optimisation, Sports Biomechanics, 22:2, 300-315, DOI: 10.1080/14763141.2022.2066015

Lin YC, Pandy MG. Three-dimensional data-tracking dynamic optimization simulations of human locomotion generated by direct collocation. Journal of biomechanics. 2017 Jul 5;59:1-8.

Alonso FJ, Cuadrado J, Lugrís U, Pintado P. A compact smoothing-differentiation and projection approach for the kinematic data consistency of biomechanical systems. Multibody System Dynamics. 2010 Jun;24:67-80.

Remy, C. D., and Thelen, D. G. (January 6, 2009). "Optimal Estimation of Dynamically Consistent Kinematics and Kinetics for Forward Dynamic Simulation of Gait." ASME. J Biomech Eng. March 2009; 131(3): 031005. https://doi.org/10.1115/1.3005148

Jackson, J. N., Hass, C. J., and Fregly, B. J. (September 16, 2015). "Residual Elimination Algorithm Enhancements to Improve Foot Motion Tracking During Forward Dynamic Simulations of Gait." ASME. J Biomech Eng. November 2015; 137(11): 111002. https://doi.org/10.1115/1.4031418

R1.3: The introduction has been updated to include a brief section that acknowledges previous efforts of researchers (some of the suggested studies above, plus some other additions from the literature) to develop approaches to reduce dynamic inconsistency in biomechanical data – alongside the approach that OpenSim offers. I have held back from reviewing this literature in detail, given the focus of this study is on the tools that OpenSim offers – with the reason being that they are widely used, and I believe much of the other work in this body of literature would be out of the reach of a number of biomechanical researchers (i.e. due to the complexity and manual implementation of modelling the multibody system). I have attempted to communicate this in the introduction and why the direction of the paper is geared towards the OpenSim tools (see lines 33-40 in revised submission with changes tracked):

The issue of dynamic inconsistency has led researchers to develop various formulaic- [2] and optimisation-based [3-6] approaches to reduce or eliminate residuals from biomechanical data. These works [2-6] have all demonstrated a capacity to reduce residuals to a minimal level, yet require substantial manual implementation of a multibody system model. OpenSim [7] is a widely used software that aims to simplify the process of modelling and simulating multibody systems to a semi-automated level. OpenSim offers the Residual Reduction Algorithm (RRA) as its main tool for minimising dynamic inconsistencies between modelled motions and external forces during gait [7].

These additional past studies (e.g. Remy & Thelen, Jackson et al.) do, however, provide good benchmarks to compare the results of the present study against – please refer to R1.4 below for a more detailed discussion on how this has been implemented.

C1.4: It would also be helpful to, in the discussion, place the results of your benchmark simulations in the context of the wider literature.

R1.4: A brief discussion benchmarking the results from the present study (particularly those of the best performing *MocoTrack* approach) has been included early in the discussion of the revised submission. Given that previous studies have used a 'residual elimination' approach to produce lower residuals than the present study – this discussion has also provided an opportunity to discuss how one might shift to a 'residual elimination' approach with *MocoTrack* (versus 'residual reduction) (see lines 284-293 in revised submission with changes tracked).

The near-zero residuals achieved by MocoTrack outperformed [9,10] or were in-line [6] with previously examined residual reduction approaches. Other studies employing 'residual elimination' strategies [3-5] have, however, been able to generate lower residuals during gait activities. To ensure an appropriate comparison to other tools in the present study, residual force and moment actuators remained in the MocoTrack approach. Given the low residuals achieved with MocoTrack, it is plausible that residual force and moment actuators could be removed to shift to a 'residual elimination' versus 'residual reduction' approach – and determine a solution with actual-zero residuals. While feasible, the impact of this altered approach on joint kinematics and kinetics, and computational time would need to be considered.

C1.5: It would be helpful to begin the methods sub sections RRA, iterative RRA, MocoTrack, and AddBiomechanics with a brief summary (1-3 sentences) of how the methods work specifically covering: what quantities the algorithm manipulates, and briefly how the manipulation is done. This will help the readers to interpret the differences that are observed in the results more easily.

R1.5: Further details around each of the processing tools have been added to the methods section in the revised submission (see 'Data Analysis' section in revised submission). There is significant detail in how these tools are described in their associated papers – particularly in Werling et al. (*AddBiomechanics*) and Dembia et al. (*MocoTrack*). I have tried my best to include the main quantities the algorithms use and how the way in which each tool approaches the residual reduction problem, while keeping this to a few brief sentences. There are subsequently details around the *AddBiomechanics* and *MocoTrack* tools (e.g. problem formulation, equations) that are not included, therefore readers are also referred to Werling et al.[a] and Dembia et al.[b] (both are open access papers) for additional details.

a. Werling K, Bianco N, Raitor M, Stingel J, Hicks J, Collins S, et al. AddBiomechanics: Automating model scaling, inverse kinematics, and inverse dynamics from human motion data through sequential optimization. bioRxiv 2023:2023–06. doi: 10.1101/2023.06.15.545116

b. Dembia CL, Bianco NA, Falisse A, Hicks JL, Delp SL. OpenSim Moco: Musculoskeletal optimal control. bioRxiv 2019:839381. doi: 10.1101/839381

C1.6: **Using these additional algorithmic details (of the previous point) you should be able to explain some of the kinematic differences that appear in the solution of AddBiomechanics, and the joint torque differences that appear in the solution of Moco. It's possible that AddBiomechanics allows this deviation to develop due to the regularization term (the third term) in Eqn. 2 of Werling et al. For Moco, its possible that these joint torques deviations could be reduced by updating some of the costs and/or switching to the foot-ground contact model.**

Obviously one can't be sure of this without doing a sensitivity analysis, which is beyond the scope of this paper. If you can, with a few hours of work, determine which parameters in AddBiomechanics and in Moco Track affect these differences it would be a useful addition to the discussion.

Werling K, Bianco NA, Raitor M, Stingel J, Hicks JL, Collins SH, Delp SL, Liu CK. AddBiomechanics: Automating model scaling, inverse kinematics, and inverse dynamics from human motion data through sequential optimization. bioRxiv. 2023 Jun 15:2023-06.

R1.6: I agree that it would be interesting to try and understand what might be driving the differences. However, as noted in your comment – a sensitivity analysis or more detailed manipulation of the underlying algorithms would be required to achieve this. There are many differences in the approaches tested, and a number of parameters or cost weights that could be altered to try and achieve this – and I feel that delving into this is outside the scope of the present study. A brief comment on the complexities and differences in the cost functions of the different approaches, and how sensitivity analyses may be useful in further understanding why/how joint kinematics differences occur has been added to the revised submission discussion (see lines 327-333 in revised submission with changes tracked):

The complexities and differences in the cost functions underpinning the different approaches tested make it difficult to pinpoint exactly why certain kinematic variations appear. Altering inputs to the various approaches (e.g. changing tracking weights in RRA; altered cost functions and weights in MocoTrack) may produce minor variations to results in the present study. Additional sensitivity analyses around these inputs are likely needed to understand the potential impact of such changes.

C1.7: **The long list of numerical results reported directly in the paper would be more easily read as a table. The figures are really useful, so its great that you've included this data graphically.**

R1.7: In line with similar comments made by reviewer #2 (see C2.3), the mean and standard deviation values originally reported in text have been removed and included in tables in the revised manuscript (see Tables 1 and 2 in revised submission).

C1.8: It is mentioned on page 8 of Dembia et al. that ground forces can be applied directly to the model (which requires the addition of non-physical residual actuors) or simulated using a foot-ground contact model. On page 7 lines 111-112 it is mentioned that the experimental external GRFs are applied directly to the model. Given that:

a. Please report the range of the forces generated by the residual actuators during the course of Moco Track's solution. This might be helpful to explain one puzzling feature of Figure 5: Moco Track's ankle torques track the data very well, but the rest of the joint torques are quite noisy.

R1.8^{a}: Apologies if this is misinterpreting what is being asked with this comment – however I believe the data in Figures 2 and 3 may already achieve this request. The individual data points for peak residual forces and moments are reported in these boxplots – and thus demonstrate the maximum range of force/torque generated by these actuators during simulations with each residual reduction approach. The associated data provided with the paper also allows readers to further inspect the residual force and moment data with each participant, gait cycle and residual reduction approach. Please clarify whether something additional or more specific to this was required – as I am happy to address as necessary.

b. If time permits, it would be very helpful to see how the solutions change when the foot-ground contact model is used in combination with an additional tracking term on the ground force trajectory.

Dembia CL, Bianco NA, Falisse A, Hicks JL, Delp SL. Opensim moco: Musculoskeletal optimal control. PLOS Computational Biology. 2020 Dec 28;16(12):e1008493.

R1.8^{b}: The use of a foot-ground contact model would be interesting to inspect, however I believe that adding this additional cost function to the *MocoTrack* approach would shift this away from an accurate comparison to the other approaches tested and the scope of this study. There seems to be two broad considerations when using a foot-ground contact model (e.g. contact spheres on the foot) in tracking simulations: (1) use generic parameters within the foot-ground contact model and solve for the adjusted kinematics that track the experimental ground reaction forces; or (2) optimise the parameters of the foot-ground contact model (e.g. sphere location and radii; stiffness etc.) to minimise the amount of kinematic deviation required to track the experimental ground reaction forces. Employing either of these approaches would certainly result in kinematic deviations compared to the original *MocoTrack* solutions – yet I believe this would mostly be the result of choice in foot-ground contact model and associated parameters (i.e. any tracking solution is highly sensitive to the foot-ground contact model). The original purpose of this paper was to more-so focus on residual reduction using experimental data – and hence why I feel venturing into the world of ground reaction force prediction is moving outside of the study scope.

C1.9: **It would be helpful to see the ground forces generated by the different methods in comparison to the recorded ground forces and CoP trajectories. While RRA applies the ground forces directly, both AddBiomechanics and Moco Track can make adjustments: AddBiomechanics makes small adjustments to compensate for force plates that are not perfectly flat (see page 27 of Werling et al.), while the ground forces from Moco Track can be solved for using a foot contact model (see page 8 of Dembia et al.).**

I suspect that some of the noisy joint torques that you're seeing in Moco's solution in Figure 5 will also be visible in the simulated ground reaction forces and CoP trajectories.

Werling K, Bianco NA, Raitor M, Stingel J, Hicks JL, Collins SH, Delp SL, Liu CK. AddBiomechanics: Automating model scaling, inverse kinematics, and inverse dynamics from human motion data through sequential optimization. bioRxiv. 2023 Jun 15:2023-06.

R1.9: As noted in R1.9^b, I believe the comparison to a foot-ground contact model within the *MocoTrack* approach is outside of the scope for the present study. Adding foot-ground contact tracking to the problem (or even tracking smooth joint kinetic signals derived from inverse dynamics) could improve the smoothness of torque signals – so while I believe testing this experimentally is outside the scope of this paper, I have included a brief comment to this in the discussion of the revised submission (see lines 370-373 in revised submission with changes tracked):

Including additional components in the MocoTrack cost function (e.g. tracking joint moments derived from inverse dynamics or using a foot-ground contact model to track and solve for external forces) could also assist in achieving smoother joint kinetic signals overall.

With respect to presenting the altered ground reaction forces from the *AddBiomechanics* approach – I am a little unsure how to access these, or if they are available for export. Within the inverse dynamics folder for each participant (see 'ID' under the 'AddBiomechanics' folder in the associated dataset), there are multiple .mot ground reaction force files included (e.g. 'Run_5_grf_raw.mot' and 'Run_5_grf.mot') – yet upon further inspection the outputs of these look identical (albeit with the headers changed). The *AddBiomechanics* viewer in the online app allows you to display additional force plate data (e.g. the original 'Force Plates' and 'Zero Residual Force Plates') – yet I am not sure whether these additional data are exported. This would be interesting to examine, however at this point I cannot action a response to this part of your comment. If it is necessary to pursue this further, I can reach out to the authors of the app for clarification.

C1.10: This is more of a comment, but the oscillatory joint torques produced by Moco in Figure 5 reminds me of the kind of output that is produced when the squared position differences are minimized on an integrated system. The classic way to deal with this problem is to minimize the squared differences of both position and velocity. The work that you are doing is quite valuable so that the defaults in Moco Track (maybe) can be improved.

R1.10: This was an interesting comment to explore and has subsequently made a large impact on the results in the paper. Initially I experimented with a single subject's data, in running a case study of the *MocoTrack* tool with and without tracking joint coordinate speeds — and this made a dramatic difference in the smoothness of the joint torques coming from the Moco solutions. In addition, considering that RRA tracks joint accelerations in its objective function, including joint speeds in the Moco objective function is likely a better comparator. These details and justifications have been included in the methods of the revised submission (see lines 133-137 in revised submission with changes tracked):

Both joint coordinate values and speeds were included in the tracking problem as initial experimentation with coordinate values only resulted in substantial noise in the joint torques of the solution, and tracking both of these quantities likely provides a better comparator to the RRAs tracking of joint coordinate accelerations.

Given this change, the *MocoTrack* simulations were re-run which resulted in some minor changes to the results. A slight increase in computational time (see line 190 in revised submission with changes tracked), and the residual force and moment values (see Tables 1 and 2) were observed. Despite the increase in residuals, these still remained the lowest of all tools and under the recommended benchmarks (see Figures 2 and 3). Perhaps the most dramatic change was in the average joint torques for the lower limb and torso becoming much smoother (see Figure 5). All updated results are reported in the revised submission.

The discussion has also been amended to acknowledge how the joint torque signals have changed with the updated simulations. Given the upper limb joint torques remained noisy, it made sense to inspect the practices around these versus the others. The main difference related to the tracking weight values used for upper limb coordinates (i.e. 1.0) versus all others (i.e. ≥ 10.0). The low tracking weights have therefore been proposed as another reason for why these noisy joint torques were present in the *MocoTrack* solutions (see lines 364-373 in revised submission with changes tracked):

The joint coordinate tracking weights used may be one explanation for the noise in MocoTrack upper limb joint torques. Tracking weights (taken from Hamner and Delp [11]) of 1.0 were used for upper limb coordinates, while all others used at least 10.0 or higher. Increasing the tracking weights for these upper limb coordinates may lead to smoother signals, but also change the balance within the objective function value *and alter convergence of the problem (e.g. more iterations with longer solution times). Including additional components in the MocoTrack cost function (e.g. tracking joint moments derived from inverse dynamics or using a foot-ground contact model to track and solve for external forces) could also assist in achieving smoother joint kinetic signals overall.*

The changes made in the methods and resulting discussion should provide guidance to future users around tracking both joint coordinate values and speeds when applying similar techniques to the present study.

C1.11: While it is nice to see some of the author's personality in the writing, you may reconsider referring to the paper as a quest. On page 25 line 332 you are not doing yourself any favours by adding 'For the sake of complete honesty, very little effort went into ...'. The work you have done is valuable but these statements may lead some readers to underappreciate your work.

R1.11: I appreciate the suggestion (perhaps I had a little too much fun writing this in the first place) – the entire manuscript has been revised and edited to remove casual language, including the specific statement mentioned here (see lines 393-395 in revised submission with changes tracked):

Most input parameters and settings used in the various approaches examined were taken from existing literature or program defaults, therefore they may not have been optimised for best performance.

C1.12: The references need some attention:

a. None of the entries has doi numbers listed.

R1.12^{a}: DOI numbers have been added to all journal article references in the revised submission.

b. If possible, replace references 4 and 10 with a referreed source. I realize this is probably not possible in the case of RRA, which to my knowledge has never (oddly) been the sole topic of a published paper. RRA is described in Chand John's PhD thesis and in Delp et al. 2007. As for reference 10, the url can be included as a note with reference 2 or as a foot note. If you decide to keep references 4 and 10 as web references you'll need to add some extra information such as the date you accessed it.

John CT. Stabilization of human walking by muscles revealed using three-dimensional muscle-driven simulations. Stanford University; 2012.

Delp SL, Anderson FC, Arnold AS, Loan P, Habib A, John CT, Guendelman E, Thelen DG. OpenSim: open-source software to create and analyze dynamic

simulations of movement. IEEE transactions on biomedical engineering. 2007 Oct 22;54(11):1940-50.

R1.12^{b}: I have opted to keep the web-based references for the RRA tool. First, I could not locate a copy of Chand John's thesis to review if this had the appropriate supporting information regarding RRA. Second, the RRA webpages were the direct source of information for what was included and come from the reputable source of the OpenSim development team. With respect to the *AddBiomechanics* web reference, this is cited when referring to where the data was processed – hence I felt it appropriate to keep this as the source where users go to input data to the server. Web-based references have been updated to include all relevant information required for the Vancouver referencing style used (e.g. date accessed).

c. Reference 5: inconsisent capitialization of the journal title

R1.12^{c}: The capitalisation of this journal title has been corrected in the revised submission.

Code

I have cloned and run (some) of the python scripts that you've put on https://github.com/aaronsfox/dynamic-consistency-quest on both a Windows machine and a Linux machine. I have a few comments:

C1.13: **On Windows I've run 01_runSimulation.py with (lines 154-157)**

```
runRRA=True
runRRA3=True
runMoco=True
runAddBiomech=True
```

and the script fails on line 532 when os.rename throws a FileExistsError.

a. If you instead use os.replace you'll have the desired effect without the FileExistsError. I've just updated os.rename with os.replace and the script runs

R1.13^{a}: All instances of *os.rename()* have been replaced with *os.replace()* in the updated code, which avoids these file exists errors.

b. until it hits the runAddBiomech code block at line 1117 and then it complains that os.chdir(f'..\\..\\..\\data\\HamnerDelp2013\\{subject}\\addBiomechanics\\{runLabel}') cannot be found. Yet if I start the script in this block by setting runRRA, runRRA3, and runMoco to False, the script appears to terminate without running addBiomechanics. Looking at the code, I can see that this is intentional from the message "This section of the script does not run the entire ...". It would be nice to see this message printed to the screen, and maybe with some short instructions on

how to proceed to use AddBiomechanics: for example, is the dataset you've loaded already available at AddBiomechanics? If not, what should I do to process your files so that I can reproduce your results?

R1.13[b]: The issue with changing directories once reaching the *AddBiomechanics* step was due to a missing line of code that returned back to the home code directory once completing the *MocoTrack* step. This has been amended in the code to avoid the error. Some additional code has been added when users set the *AddBiomechanics* pipeline to run, whereby an output is printed to the terminal that alerts users to what is happening in that step (i.e. that the data needs to be uploaded to the server for these analyses to be re-run). In these outputs, users are prompted to contact the code authors if the instructions are unclear. It is hoped that this direct message printing will help users understand what is occurring in this step.

c. The scripts 02_collateSimulations.py and 03_analyseSimulations.py ran smoothly. When I looked at results/HamnerDelpDataset/outputs I was happy to see that the results had been updated. However, when I looked at results/HamnerDelpDataset/figures the figures had not been updated. It would be best to update the script so that the figures are updated automatically.

If I manually deleted the existing figures then new ones were generated, and these new plots matched the published versions with slight differences in timing (due to the machine). Some of the line colours in residualForces.png, and residualMoments.png looked odd (for example the AddBiomechanics box and whisker plots had blue whiskers) but this migth be a problem with my python installation: it was giving a 'FutureWarning' that passing 'palette' without assigning 'hue' is deprecated ... along with a few other warnings.

R1.13[c]: In response to this comment and to hopefully make running the code altogether easier, the original 3 Python scripts have been combined into one singular script – now *runSimulations.py*. I wanted to leave the option open for users to run different parts of the script (e.g. simulations, compiling data, generating figures etc.), and hence have still kept these unlinked (i.e. you could theoretically run the simulations without updating the figures). However, the main README has been updated to explicitly alert users to set the *compileData* and *analyseData* flags to **True** to ensure that accurate up-to-date outputs are generated.

I have updated how the figures are recreated to use base Matplotlib in generating the boxplots (instead of Seaborn). I have encountered the issues you describe before with Seaborn boxplots when using different versions of this package. I have also updated the repository README to include specific details of the Python package versions used with this code. These two aspects combined should fix any issues associated with reproducing the figures.

d. If possible, please make sure that someone can run all of your scripts from scratch (with new simulations) without error.

R1.13[d]: Thank you for this suggestion. After making the above (and other) updates to the code I was able to test it on another Windows machine in our lab without error. Hopefully this carries across to other users.

C1.14: As it turns out, may main work computer is a Linux machine, and so, I've also tested 01_runSimulations.py on my Linux machine. There are a number of path related problems that should be fixed if you want to be cross-platform compatible:

a. Every path that appears in your python scripts needs to be updated using the os.path.join command. For example:

Near line 178 of 01_run Simulations.py appears a path that is Windows specific:

With open(f'..\\..\\data\\HamnerDelp2013\\{subject}\\expData\\gaitTimes.pkl', 'rb') as openFile:

Replacing this line with the following allows it to run on any machine:

with open(os.path.join('..','..','data','HamnerDelp2013',subject,'expData','gaitTimes.pkl'), 'rb') as openFile:

b. The XML files that OpenSim generates for RRA, for example, have windows-specific slashes. For example in subject01_run5_cycle1_setupRRA.xml the model file is set as

..\..\model\subject01_adjusted_scaled.osim

which is windows-specific. Changing these path slashes

../../model/subject01_adjusted_scaled.osim

allows the model to be loaded. I think that this can be done by using the os.path.join command in, for example, the rraTool.setModelFilename(...) command (and all similar commands).

R1.14: Thank you for this recommendation — I believe this will be useful to me for this project plus any in the future where I share analysis code. I have replaced the paths in all scripts to be constructed using *os.path.joint()* which should hopefully address cross-platform issues. Unfortunately, I am not adept to or have access to a Linux environment and hence was unable to test this. However, I have added details to the datasets main README (see https://zenodo.org/records/10347733) about code being tested only on a Windows machine and the associated Python dependencies.

C1.15: The script assumes that OpenSim is installed at 'C:\\OpenSim 4.3\\Geometry'. It would be very helpful if, perhaps in the README.md file, the details of the installation and dependencies on your machine where listed. In addition, it would be useful if some hint was provided about which paths need to be updated in the code if the installation layout and/or operating system differs from the one that you used.

R1.15: In response to this comment, details have been added to the datasets main README (see https://zenodo.org/records/10347733) specifying the assumption that the OpenSim geometry install path is at *C:\OpenSim 4.3 Geometry*. The analysis script (see *runSimulations.py*) also now prints out a message about where the geometry path has been set and prompts users to change this if incorrect.

C1.16: There are likely many other small things to correct to get the code you've written functioning cross-platform. Getting code to run flawlessly cross platform is tedious work, but it will allow a lot of other people to use your code. I see two ways to proceed:

a. Get access to a Linux machine (or a colleague with a Linux machine) and do the work to get everything functioning on Linux. If you have never used Linux, it would be best to get help with this - there are a lot of small things that will be time consuming for you to learn on your own.

b. Explicitly mention that the code is developed for Windows, but in principle, can be made to work cross-platform with the edits that I've mentioned. The code is publicly available, and so, in the best case someone who'd like to use your work will contribute the updates needed to make the code run cross-platform.

R1.16: As noted in R1.14, I am unable to confirm proper functioning of code on a non-Windows system, and hence added details to the datasets main README (see https://zenodo.org/records/10347733) about code being tested only on a Windows machine and the associated Python dependencies.

C1.17: While it is convenient to put binary data in a github repository, this can cause problems: if the data ever changes git will store just store, basically, the old file and the new file. In the case of simulation data this will make the repository size balloon. Usually data is stored separately from git, and not under revision control. You could for, example, store the non-python code contents of data/HamnerDelp2013, paper, preprint, and results folders in Zenodo.

R1.17: I noted earlier that reviewer #1 accessed the associated code and data directly from the GitHub page – however the desired approach for eventual readers is to use the SimTK link (i.e. https://simtk.org/projects/dynamic-quest) specified in the *Data Analysis* section of the paper. The SimTK page directs to a Zenodo link (see https://zenodo.org/records/10347733) which will archive the code and data in the current state at paper acceptance. It is possible that some updates may occur on the

GitHub page, but these will be unaffected by readers visiting the SimTK page and linked Zenodo archive. Theoretically this should provide a stable, consistent and working analysis code for any readers of the paper.

Reviewer #2

C2.1: **While the framing of the study as a "Quest" makes the paper more interesting for the reader, the language at times crosses the line into an overly casual tone. Specifically, it is suggested to revise the following lines:**

- Line 55: "nay, quest"
- Line 332: "for the sake of complete honesty, very little effort…"

R2.1: In line with this suggestion also made by reviewer #1 (see C1.11 and R1.11) – the entire manuscript has been revised and edited to remove casual language, including the two statements mentioned here.

C2.2: **Lines 43 and 44 are somewhat misleading. Specific recommendations do exist for RRA. The OpenSim documentation provides guidelines for acceptable kinematic error and residual forces/moments, Hicks et al 2015 provide an extensive discussion on the topic, and references [5] (Samaan et al) and [6] (Sturdy et al) in the present manuscript all provide suggestions on acceptable thresholds and how many iterations should be required. For purposes of the present manuscript, it would be more accurate to state that there are not yet guidelines for selecting between RRA and other new approaches, such as Moco and AddBiomechanics.**

R2.2: I believe there was a lack of clarity around the language originally used in this statement which I have attempted to clarify in the revised submission. It's important to note that this is referring to repeat iterations of the same RRA process (i.e. using the same task weights on an updated model and kinematics) – which differs to the iterative process of the studies by Samaan et al. and Sturdy et al. (i.e. iteratively adjusting task weights to optimise and achieve minimised residuals). The statement in the revised submission has been edited to include the fact that benchmarks for residuals (i.e. from Hicks et al.) exist – however it is difficult to determine whether a singular or iterative RRA approach can achieve these benchmarks. Further, a statement has been added to highlight the lack of research benchmarking the RRA to both existing and new approaches for residual reduction (see lines 46-51 in revised submission with changes tracked):

While benchmarks exist for acceptable levels of residuals [1], there are no guidelines on whether these benchmarks are achievable with a single RRA iteration, or whether a set number of RRA iterations is required. Lastly, there are no studies determining whether OpenSim's RRA tool offers comparable residual reduction to other approaches in the literature [2-6] or whether new tools [13,14] may outperform this.

C2.3: **The Results section would benefit greatly from a table that compares the mean±SD values which are currently included in the text. As written, it is difficult for the reader to quickly compare the various methods. It would particularly help highlight the tradeoff in computational time and residual forces/moments and kinematic error.**

R2.3: In line with similar comments made by reviewer #1 (see C1.7), the mean and standard deviation values originally reported in text have been removed and included in tables in the revised manuscript (see Tables 1 and 2 in revised submission). I have, however, left the in-text reporting for computational time — as this would be reported in a single-line table and potentially not offer great benefit over the figure combined with the in-text reporting.

C2.4: The plots (Figures 1-5, 7) are impossible to read in grayscale (and difficult to read even in color). Distinct symbols and different line styles should be used to ensure readability without color. Legends should be added to the figures.

R2.4: Thank you for this feedback, and upon reviewing these in grayscale it is clear they were challenging to review. In an attempt to address this issue, each tool has been attributed a different marker and line-style combination across all relevant figures. Where appropriate, these marker/line-style notations have also been identified in a figure legend as well as the figure caption. In making these changes to the joint kinematic and kinetic figures, it became clear that the error clouds around each line visualising the standard deviation about the mean needed to be removed (i.e. with the error clouds the maker/line-style differences were still not clear) – and hence this update has also been made to the figures (see Figures 4 and 5). Despite these changes, I still believe there may be some difficulties in distinguishing the data from the different approaches (particularly in joint kinematics and kinetics) – however this seems unavoidable given the significant similarities/overlap in some of the data being presented. I am hopeful that the changes made has assisted in improving the data visualisation in the paper – but am open to any further suggestions to assist with this.

C2.5: Line 267 states that MocoTrack substantially outperforms previous efforts in [5] (Samaan et al) and [6] (Sturdy et al). The optimizations tested by Samaan et al required 64 and 98 hours to complete, which is indeed a substantial difference. Sturdy et al tested 200 iterations which required ~2 hrs to complete. However, most of the improvements for the 5 m/s case occurred between 30-50 iterations, which would correspond to approximately 20-30 minutes. This is comparable to the MocoTrack results. It seems that a comprehensive investigation of the available options should consider this approach, as it is unclear how the results from 30-50 "optimized" RRA iterations would compare to MocoTrack.

R2.5: Thank you for flagging these additional findings, particularly from the work of Sturdy et al. I believe testing of the algorithm employed by Sturdy et al. is outside the scope of this study (or more so would simply be repeating their original analyses) — however a more balanced discussion of the computational time for *MocoTrack* in the present study compared to this existing literature has been included in the revised submission (see lines 295-209 in revised submission with changes tracked).

While the MocoTrack approach produced lower residuals, it took approximately 4-60 times longer than all others on average. Although MocoTrack had the highest computational times in the present study, the approximate 15-30 minute time-range is *not overly burdensome compared to previous optimisation-based approaches [3] or those attempting to optimise the RRA process [9,10]. Both Samaan et al. [9] and Sturdy et al. [10] ran optimisations to select joint coordinate tracking weights that best minimised residual forces and moments. Samaan et al. [9] tested particle swarm and simulated annealing algorithms, which had average convergence times of 64.6 and 98.1 hours, respectively — substantially greater than any computational approach in the present study. The tracking weight selection algorithm tested in Sturdy et al. [10] was much faster, identifying tracking weights that achieved residuals below recommended thresholds in 200 RRA iterations at ~2-hours. While the 2-hours exceeds that of MocoTrack in the present study, residuals were substantially reduced in the first 50 RRA iterations [10] — which would likely correspond to ~30-minutes of computational time (i.e. aligning with MocoTrack).*

C2.6: Figure 6 is a nice visual comparison of the various methods. However, RRA commonly allows substantial pelvis translational drift (particularly in the vertical and anterior/posterior directions) in order to reduce the residual forces. It appears that in Figure 6 pelvis translational error is not represented. This is ok, as it allows visual comparison of other kinematic differences, but no mention is made about pelvis translational error, and it is omitted Figure 4. Results for pelvis translational error should be reported.

R2.6: Not including pelvis translations in the original kinematic figure was an oversight – so thank you for identifying this error. These data have now been included with the in the figure displaying mean kinematics from each tool (see Figure 4). Figure 6 in the original submission (and maintained in the revised submission) did, however, already consider pelvis translations in displaying the average joint motions from each tool. The updated mean kinematics in Figure 4 demonstrate the similarity in pelvic translation data across the tools – and hence why it seemed like they had been excluded. The additional pelvis translation data now included in Figure 4 will now help readers identify the subtle differences in Figure 6 (e.g. the slightly more posteriorly translated pelvis from *AddBiomechanics* results).

C2.7: The noise in joint kinetics from MocoTrack warrants further discussion. A common use of RRA is as a preprocessing step before a muscle-driven simulation. The oscillations in joint moments in the MocoTrack results would lead to extremely poor muscle-driven simulation performance. The deeper question here is – are the noisy joint moments more “true”? The author mentions soft tissue motion as a potential source of the oscillation. However, does that mean that the joint moments are in fact oscillatory? Biomechanists tend to prefer smooth joint moments as input to a muscle-driven simulation. However, if the joint moments are in fact oscillatory due to soft tissue motion, that represents a significant claim.

R2.7: As a broad initial response to this comment, I believe it would be incredibly difficult to understand prove/disprove that the noisy joint moments are more or less ‘true.’ Without a gold standard criterion measure for these (which this dataset does not have), there would be no way to truly answer this.

In response to a comment from reviewer #1, it appeared that a lot of the noise in the joint kinetic signals could be resolved by tracking both the joint positions and speeds (see C1.10 and R1.10 for specific details). This change largely smoothed out the joint kinetic signals for lower limb coordinates, however some noise remained in the upper limb kinetic signals. Additional discussion around this has now been included in the revised submission (see lines 364-373 in revised submission with changes tracked and R1.10).

The notion that noisy joint moments would likely lead to poor muscle-driven simulation performance has also been included at this point of the discussion in the revised submission. Readers are made aware that some manipulation of the approaches used in the present study may be required if progressing to muscle-driven simulations (see lines 383-387 in revised submission with changes tracked):

Irrespective of the mechanism, oscillatory joint moments such as those observed would likely lead to poor muscle-driven simulation performance. Studies progressing to muscle-driven simulations may benefit from manipulating the joint coordinate tracking weights to avoid noisy signals, or smoothing joint torque signals as a pre-processing step.

C2.8: Line 123: Typo, “participants” should be “participant’s”

R2.8: This typo, alongside a number of similar typos related to “participant’s” (e.g. in figure captions) have been corrected in the revised manuscript.

C2.9: Line 396: The last sentence in the acknowledgments appears to have been cut off

R2.9: Thank you for pointing this out. This has been addressed in the revised submission.