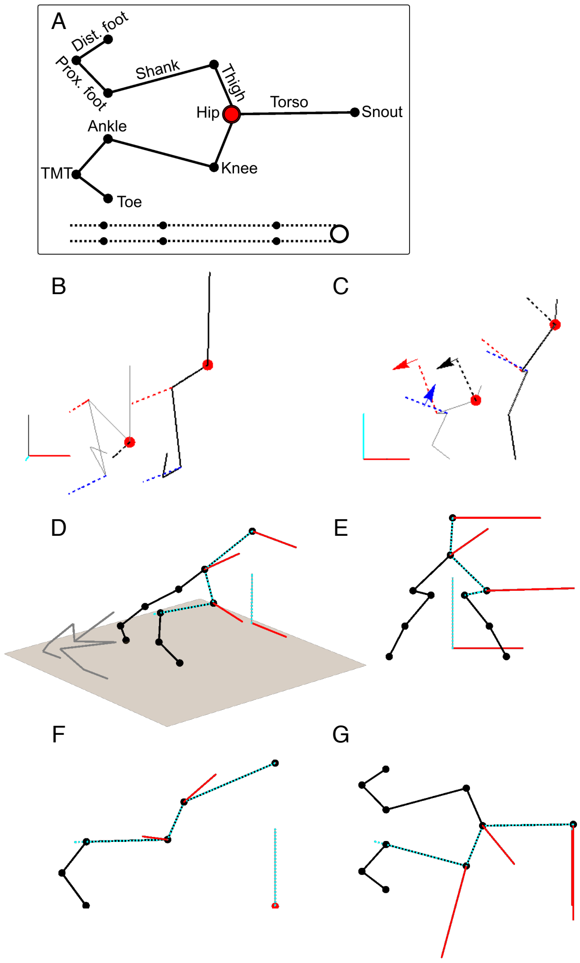

Figure 1. Schematic view of a frog jump. (A) Inset showing top view (XY plane) with body segments and joints in a natural configuration. Dashed lines extending back from the centre of mass (red circle) show leg segments in the "zero" position using null quaternions. (B) An exemplar jump in top view and (C) rear view showing the first (gray) and final (black) frames of the left leg. Dashed lines show axes of rotation for the hip (black), knee (red) and ankle (blue) highlighting how their orientations change in the direction of the arrows to align throughout the jump (see also SI Movie 1) (D) Initial limb (gray) and end−jump configurations (black) on a floor (grey square) representing the global XY plane. Local reference frames are shown with the local X (red) and Z (cyan dashed) in all frames (global frame, snout frame, torso frame, hip frame) to illustrate reference frame transformation using "quaternionization". Local Z−axes, by definition, align along each body segment. Y−axes as well as ankle and TMT reference frames have been omitted for clarity. (E) End−jump, rear view, (F) side view and (G) top view.

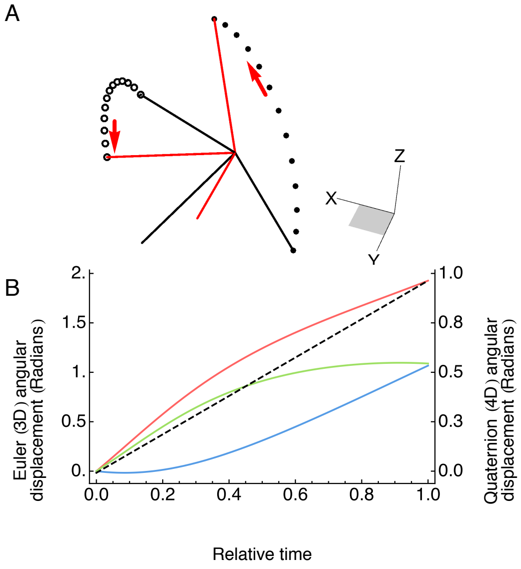

Figure 2. Calculating a smooth path of minimal rotation between two orientations is challenging in 3D, but trivial in 4D. (A) A Cartesian reference frame (Black) is rotated to a new orientation (Red) along a path determined by Spherical Linear Interpolation (SLERP) to calculate the minimum rotation required. Paths of the X axis (open circles) and the Z axis (closed circles) illustrate the smooth curved motion between the two orientations. (B) The rotation shown in (A) is parameterized in 3D space (Euler angles) versus 4D space (quaternions, dashed). Euler angles about X (Red) followed by Y (Blue) then Z (Green) axes are unpredictable and nonlinear and thus are difficult to extrapolate meaningfully. However, quaternion displacement via SLERP is linear (i.e. a great arc on the hypersphere surface), making extrapolation trivial.

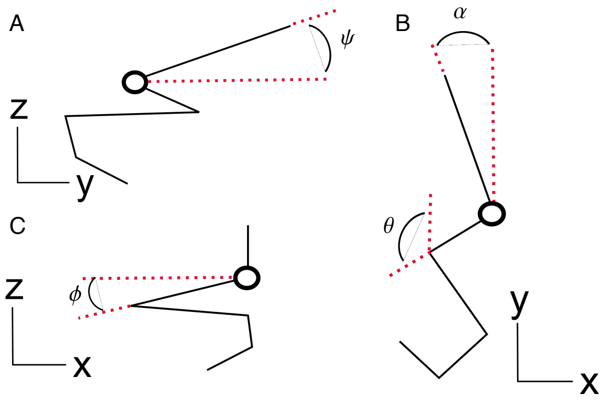

Figure 3. Segment angle definitions. Schematic of left leg in (A) side view showing the target pitch angle with respect to horizontal (B) top view showing the target yaw angle and the retraction angle of the thigh with respect to the global y axis and (C) rear view showing the adduction angle with respect to horizontal. The black circle marks the hip. Note that segment angles (pitch, yaw, retraction, adduction) are all calculated in the global reference frame (as opposed to segment quaternions which are in local reference frames; see text).

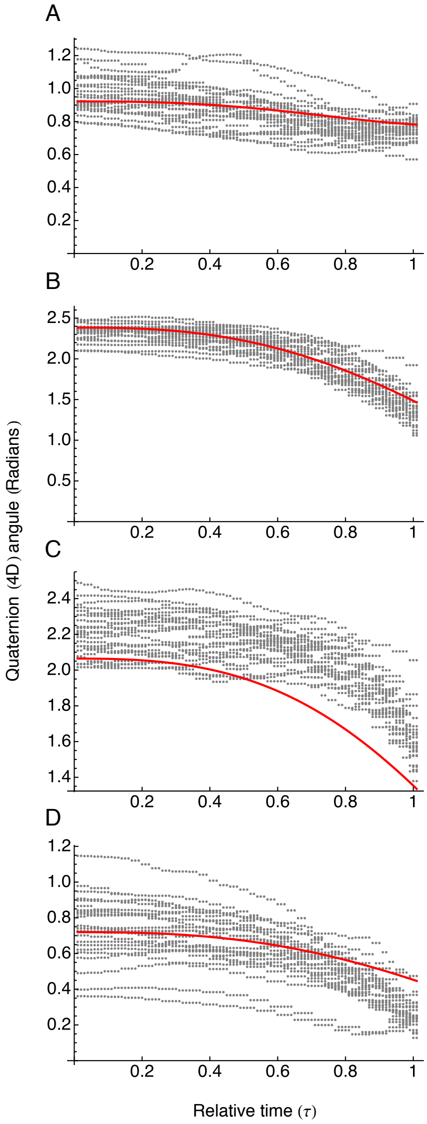

Figure 4. Experimental versus simulated frog jump kinematics. Scalar angles (in 4D) were calculated from quaternion unit vectors for body segments of (A) the thigh, (B) shank, (C) proximal foot and (D) distal foot compared to the unrotated "ground" reference frame. Grey dots are experimentally collected data points from 24 intermediate−height jumps (Richards et al., 2017). Red lines indicate kinematics simulated using SLERP. In the present convention, 4D angles of 0 would indicate that segments are at the "zero" (null) position (i.e. leg segments straightened caudally from the hip; Fig. 1 inset). Large angle values indicate a large deviation from the null configuration of the limb. In a frog jump, the limb joints extend to straighten the limb. Thus, the angle values decrease rather than increase through time as the entire limb extends.

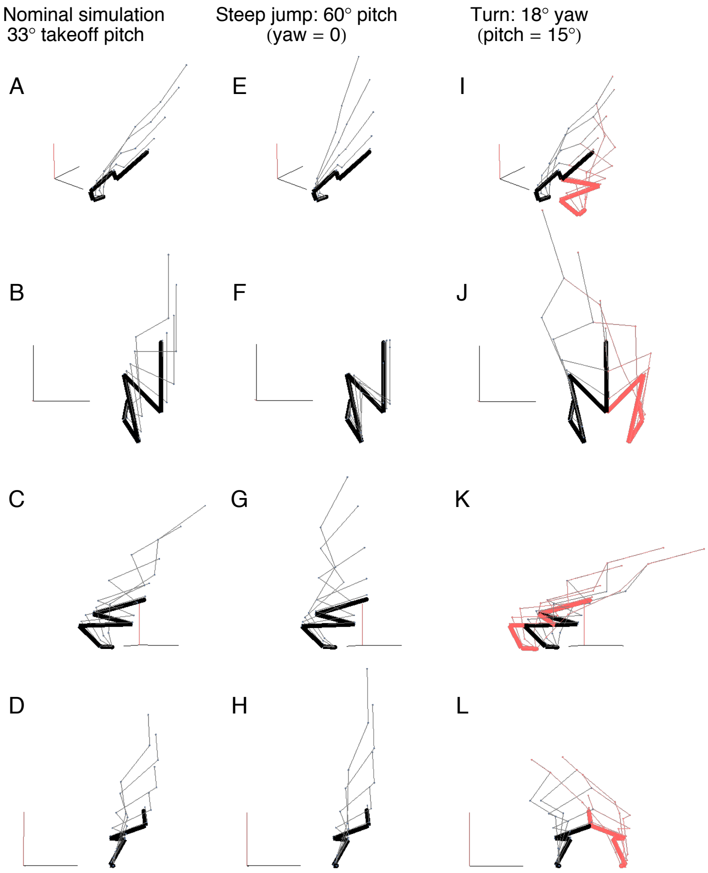

Figure 5. Simulated kinematics animations. Jump animations are shown for (A–D) nominal takeoff kinematics, (E–H) a steep jump, (I–L) a left turn shown in angled view (row 1), top view (row 2), side view (row 3) and rear view (row 4). A fixed subset of evenly–spaced animation frames are shown in each view. For the non–turning jumps (A–H) only the left leg is shown (Black), whereas both left and right (Red) legs are shown for the turning simulation (I–L). Note that initial configurations (Bold lines) are identical for each condition. The x and y axes of the global coordinate frame are shown in black and the z–axis is red.

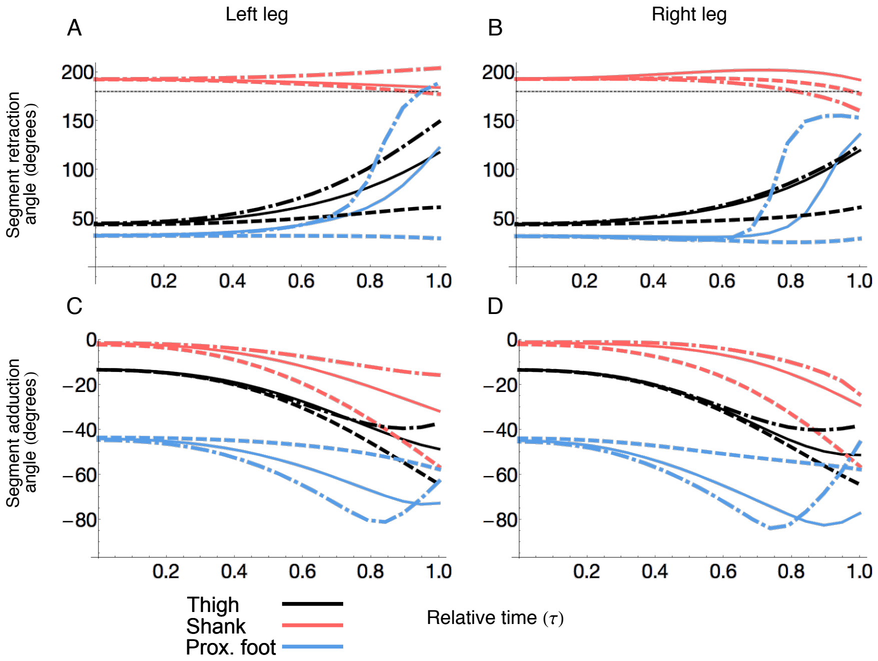

Figure 6. Simulated limb segment protraction−retraction and abduction−adduction angles for varying jump steepness and turning. Traces are for thigh (black), shank (red) and proximal foot (blue) for (A,B) retraction in the cranio−caudal direction and (C,D) adduction in the dorso−ventral direction shown for both left and right legs during a nominal jump (solid), a steep jump (dashed) and a turn (dotted−dashed) as in Fig. 5. The dashed line (A,B) represents a line drawn posterior from the hip joint from which protraction−retraction angles were referenced. Trending towards the line denotes segment caudal rotation (retraction) to push the body forward. The x−axis (C,D) represents the horizontal axis. Downward slopes indicate downward rotation (adduction) to push the body upwards. Kinematics of the distal foot are similar to the pattern for the proximal foot and therefore have been omitted for clarity.

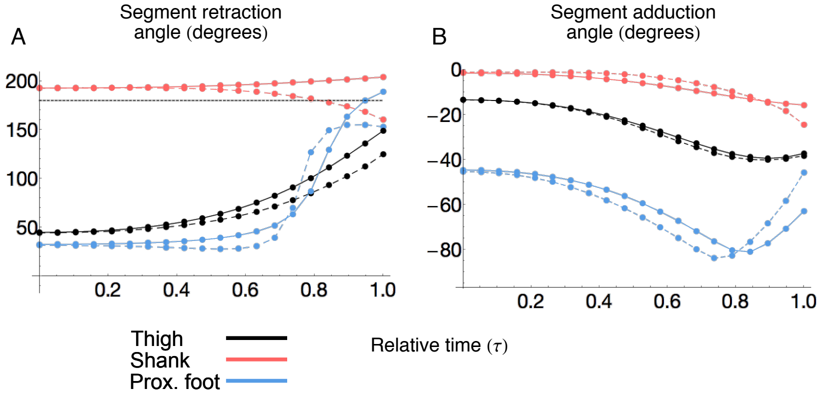

Figure 7. Left versus right limb kinematics for a left turn. Data traces are from Fig. 6, but rearranged to highlight left−right asymmetries. Traces are for left leg (solid) and right leg (dashed) using the same colours as in Fig. 6. (A) Retraction in the cranio−caudal direction and (B) adduction in the dorso−ventral direction. Note in (A) how the right shank trends downward towards the dashed line indicating retraction to push the limb forward versus the left shank which trends upwards (protracts) to push the limb backwards on the inside of the turn. Kinematics of the distal foot are similar to the pattern for the proximal foot and therefore have been omitted for clarity.

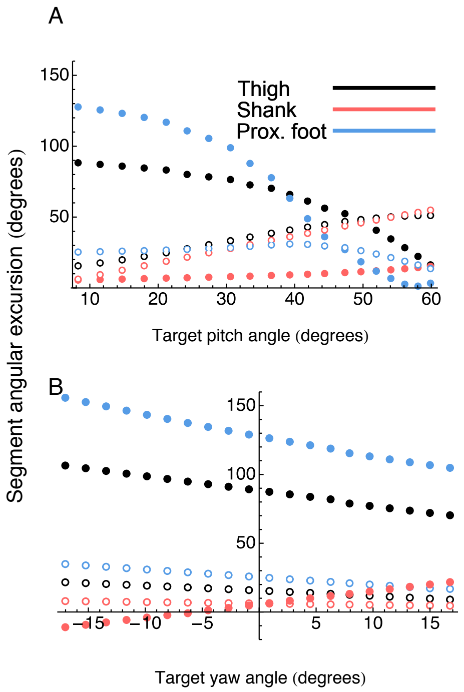

Figure 8. Left leg angular excursion for varying jump steepness and turning. (A) Varying pitch angle relative to horizontal (yaw = 0). Total retraction excursion (max retraction angle − min retraction angle [closed circles]) and adduction excursion (max adduction angle − min adduction angle [open circles]) of the thigh (black), shank (red) and proximal foot (blue). Each data point represents a single simulation beginning from the nominal initial limb configuration and ending at the specified target angle. Note that increasing jump steepness requires increased thigh and shank adduction while retraction decreases. (B) Varying turn angle (constant takeoff pitch = ~8 deg). Turns range from left (negative values) to right (positive). Negative excursion values indicate protraction. Note that unlike varying pitch, changes in the magnitude of retraction modulate turn angle.

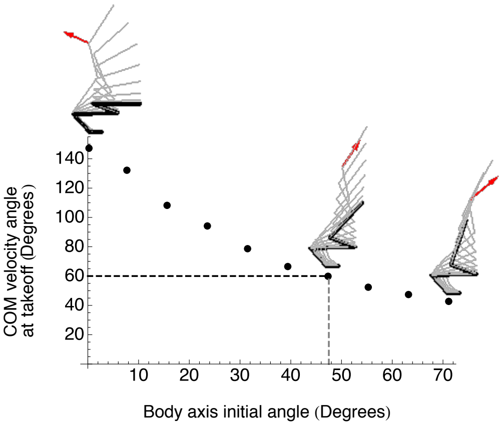

Figure 9. The effect of initial angle on jump trajectory and centre of mass (COM) takeoff velocity. Each point represents a single simulation whose entire kinematic path is influenced by initial angle. The initial pitch angle of the torso segment was varied from 0 (horizontal) to nearly vertical, leaving the leg segments unchanged. The final pitch of the body axis was held at 60 degrees for all simulations. The dashed black lines represent the optimal initial pitch angle which allows the COM takeoff velocity to align with the takeoff body orientation (i.e. the frog COM will travel in the appropriate direction). Stick figure animations for minimum, maximum and optimum initial angles show the initial posture (bold) and subsequent animation frames (gray). Red arrows indicate the direction of the takeoff velocity vector. Note that as the initial pitch angle increases, takeoff velocity direction shifts from >90 (jumping upwards and backwards) to <90 (upwards and forwards).

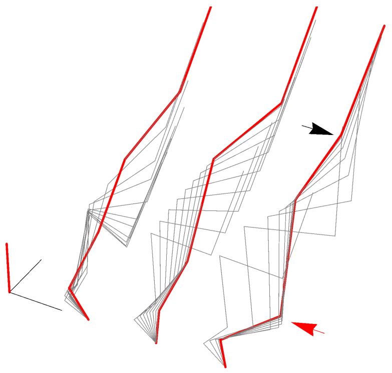

Figure S1. Euler angle−based inverse kinematics (IK) versus quaternion−based kinematics (limbSLERP). Kinematics of the left leg and body are shown in grey with the final posture at takeoff in red. Experimental observations (left) are compared with limbSLERP (middle) versus IK (right). Global XYZ axes are shown with Z in red. limbSLERP predicts jump behaviour reasonably well, however IK is unreliable for frog jumps. Note the unnatural hyperextension of the hip (black arrow) and, more problematically, the migration of the ankle joint to the opposite side of the frog (red arrow). We also note that in our implementation using *Mathematica* on OSX, IK was ~35x slower to compute.