# EECS 367 Lab

## KinEval Pose Parameters and HTML5 Audio

## Administrative

Assignment 3: Forward Kinematics and Assignment 4: Dance Controller are now **both** due at 11:59pm on **Friday, October 30**

Quiz 3 remains on **Wednesday, October 28**

# Lab Takeaways

1. KinEval overview
2. KinEval walkthrough
3. Adding music for your dance

→ How to start Assignment 4

## Dance Controller Overview

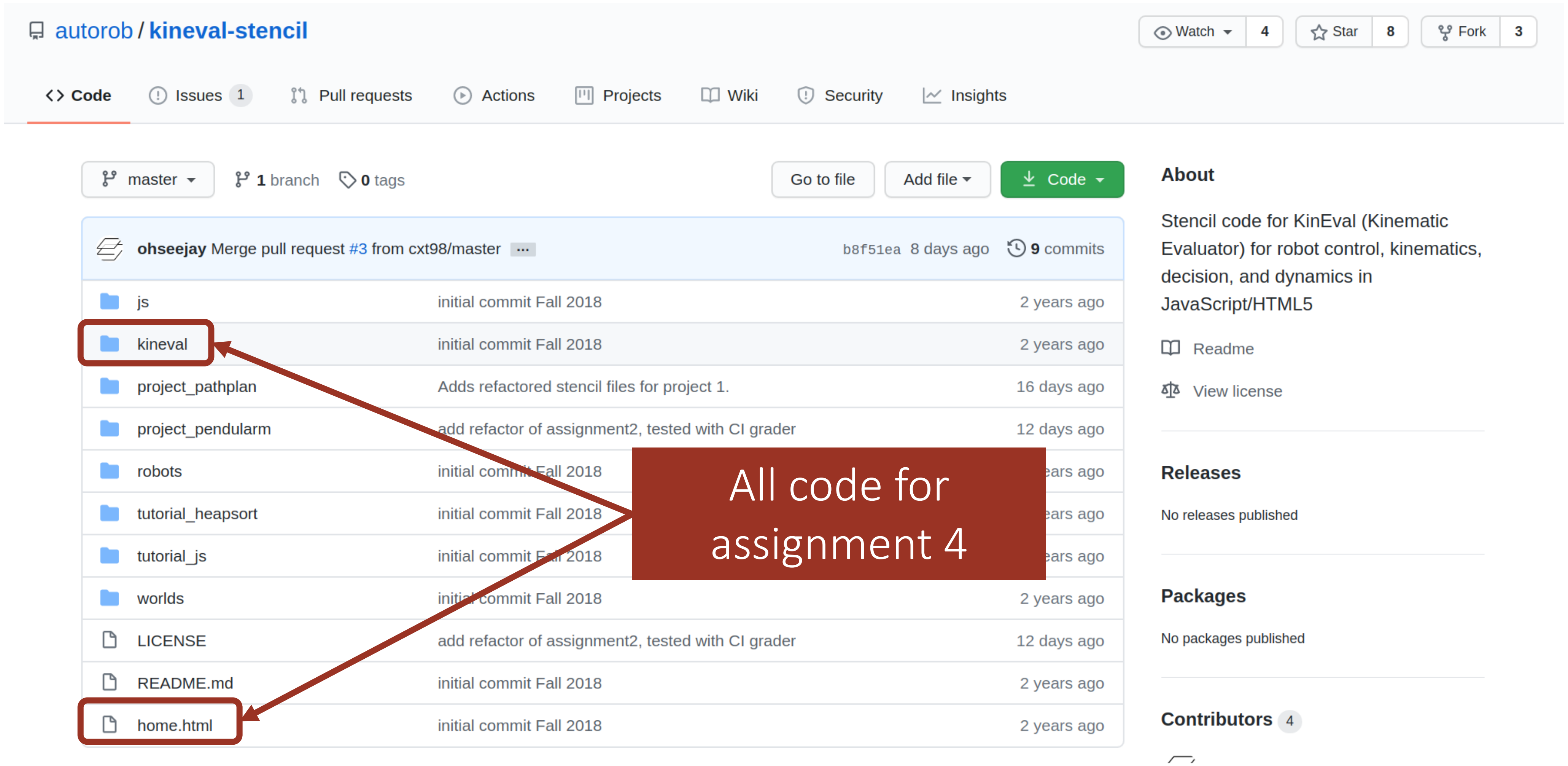
| Assignment 4: Dance Controller | | |
|---|---|---|
| 6 | All | Quaternion joint rotation |
| 2 | All | Interactive base control |
| 2 | All | Pose setpoint controller |
| 2 | All | Dance FSM |
| 2 | Grad | Joint limits |
| 2 | Grad | Prismatic joints |
| 2 | ~~Grad~~ | ~~Fetch rosbridge interface~~ |

Quaternion joint rotation, Interactive base control, Pose setpoint controller, Dance FSM: Features assigned to all sections

Joint limits, Prismatic joints: Features assigned to grad section only

Fetch rosbridge interface → Cancelled due to COVID-19

## KinEval Overview



autorob / kineval-stencil

Watch 4 | Star 8 | Fork 3

Code | Issues 1 | Pull requests | Actions | Projects | Wiki | Security | Insights

master | 1 branch | 0 tags | Go to file | Add file | Code

ohseejay Merge pull request #3 from cxt98/master ... b8f51ea 8 days ago 9 commits

| | | |
|---|---|---|
| js | initial commit Fall 2018 | 2 years ago |
| kineval | initial commit Fall 2018 | 2 years ago |
| project_pathplan | Adds refactored stencil files for project 1. | 16 days ago |
| project_pendularm | add refactor of assignment2, tested with CI grader | 12 days ago |
| robots | initial commit Fall 2018 | ears ago |
| tutorial_heapsort | initial commit Fall 2018 | ears ago |
| tutorial_js | initial commit Fall 2018 | ears ago |
| worlds | initial commit Fall 2018 | 2 years ago |
| LICENSE | add refactor of assignment2, tested with CI grader | 12 days ago |
| README.md | initial commit Fall 2018 | 2 years ago |
| home.html | initial commit Fall 2018 | 2 years ago |

**About**

Stencil code for KinEval (Kinematic Evaluator) for robot control, kinematics, decision, and dynamics in JavaScript/HTML5

Readme

View license

**Releases**

No releases published

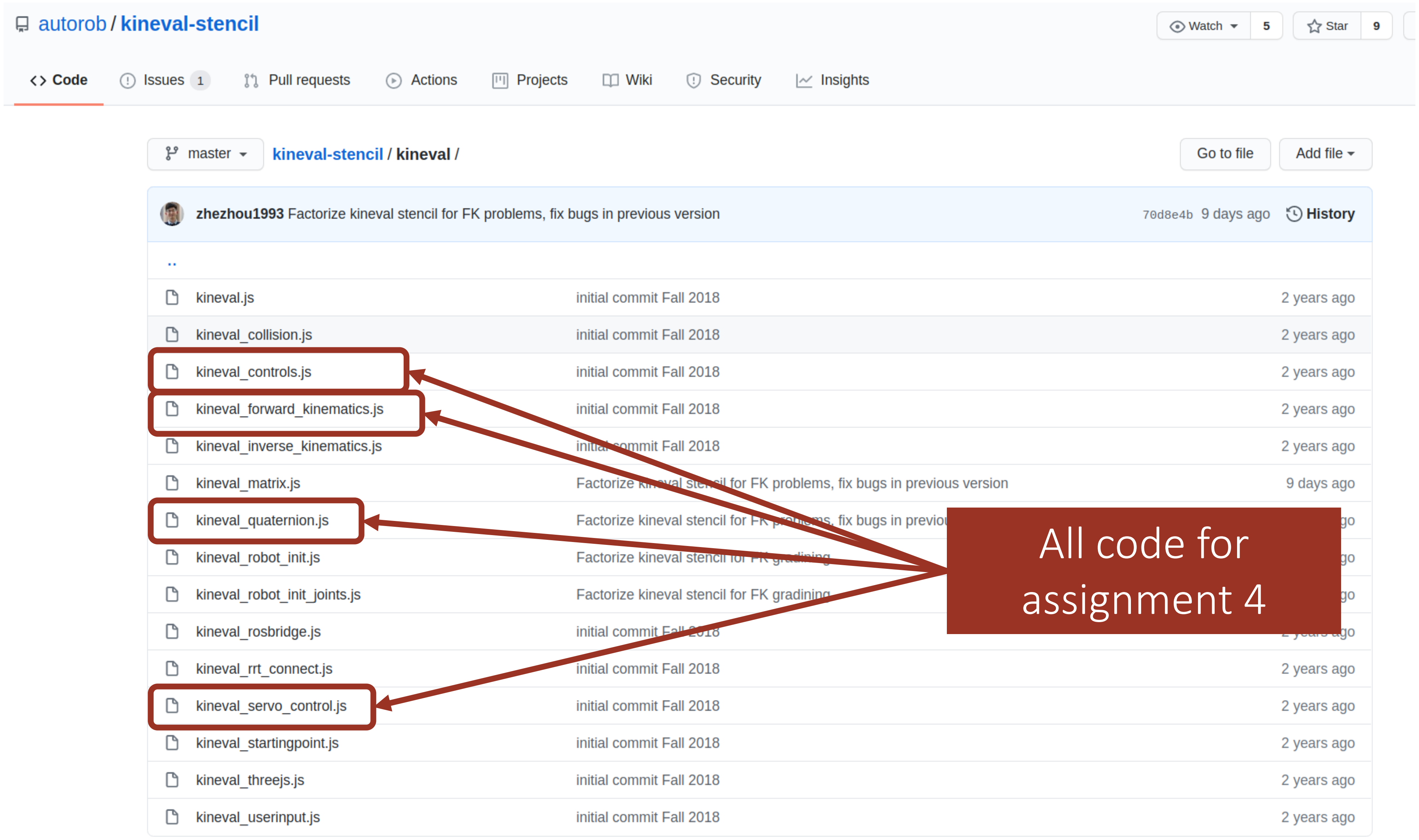
**Packages**

No packages published

**Contributors** 4

All code for assignment 4

# KinEval Overview



autorob / kineval-stencil

Watch 5 Star 9

Code | Issues 1 | Pull requests | Actions | Projects | Wiki | Security | Insights

master kineval-stencil / kineval /

Go to file | Add file

zhezhou1993 Factorize kineval stencil for FK problems, fix bugs in previous version 70d8e4b 9 days ago History

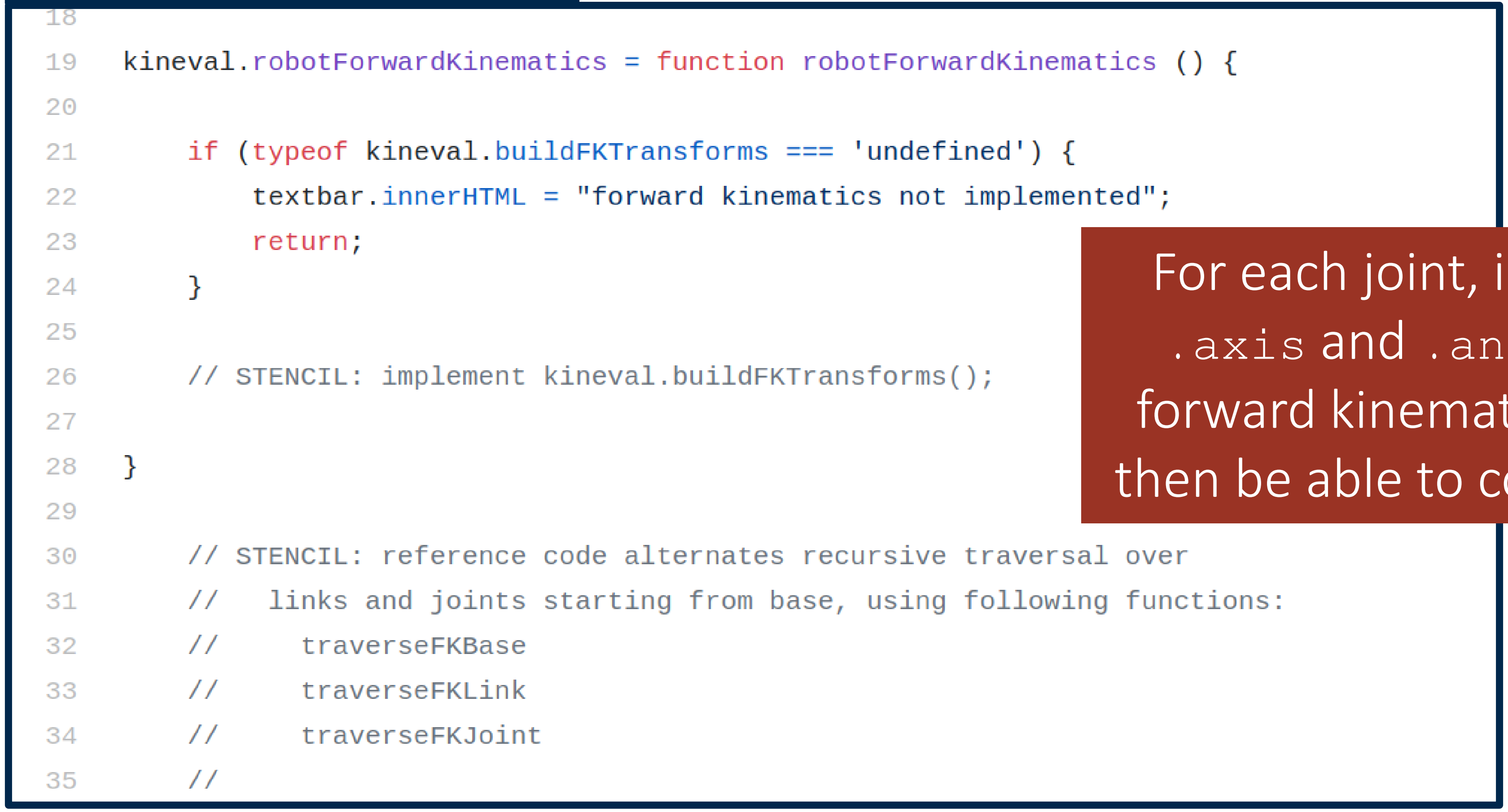
..

| | | |
|---|---|---|
| kineval.js | initial commit Fall 2018 | 2 years ago |
| kineval_collision.js | initial commit Fall 2018 | 2 years ago |
| kineval_controls.js | initial commit Fall 2018 | 2 years ago |
| kineval_forward_kinematics.js | initial commit Fall 2018 | 2 years ago |
| kineval_inverse_kinematics.js | initial commit Fall 2018 | 2 years ago |
| kineval_matrix.js | Factorize kineval stencil for FK problems, fix bugs in previous version | 9 days ago |
| kineval_quaternion.js | Factorize kineval stencil for FK problems, fix bugs in previou | go |
| kineval_robot_init.js | Factorize kineval stencil for FK gradining | go |
| kineval_robot_init_joints.js | Factorize kineval stencil for FK gradining | go |
| kineval_rosbridge.js | initial commit Fall 2018 | 2 years ago |
| kineval_rrt_connect.js | initial commit Fall 2018 | 2 years ago |
| kineval_servo_control.js | initial commit Fall 2018 | 2 years ago |
| kineval_startingpoint.js | initial commit Fall 2018 | 2 years ago |
| kineval_threejs.js | initial commit Fall 2018 | 2 years ago |
| kineval_userinput.js | initial commit Fall 2018 | 2 years ago |

All code for assignment 4

## kineval_forward_kinematics.js Revisited

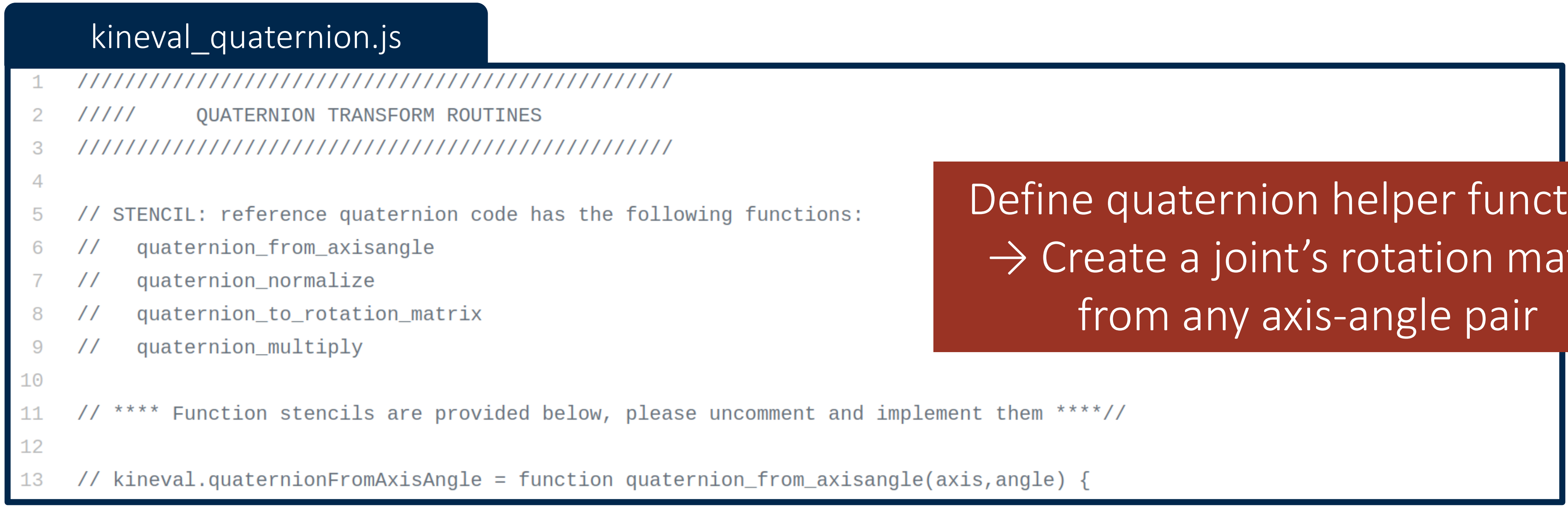
kineval_forward_kinematics.js



```
kineval.robotForwardKinematics = function robotForwardKinematics () {

    if (typeof kineval.buildFKTransforms === 'undefined') {
        textbar.innerHTML = "forward kinematics not implemented";
        return;
    }

    // STENCIL: implement kineval.buildFKTransforms();

}

    // STENCIL: reference code alternates recursive traversal over
    //   links and joints starting from base, using following functions:
    //     traverseFKBase
    //     traverseFKLink
    //     traverseFKJoint
    //
```

For each joint, incorporate `.axis` and `.angle` within forward kinematics. You will then be able to control joints!

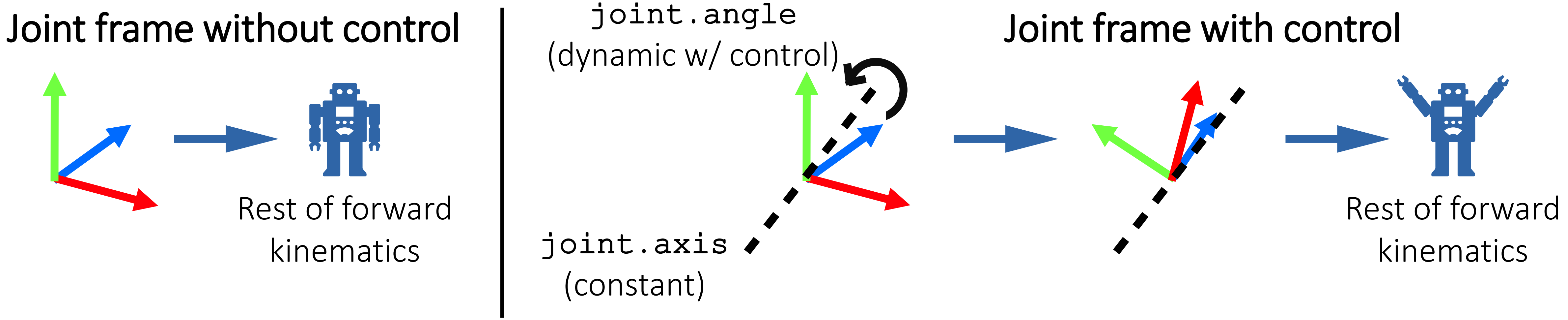
# kineval_quaternion.js



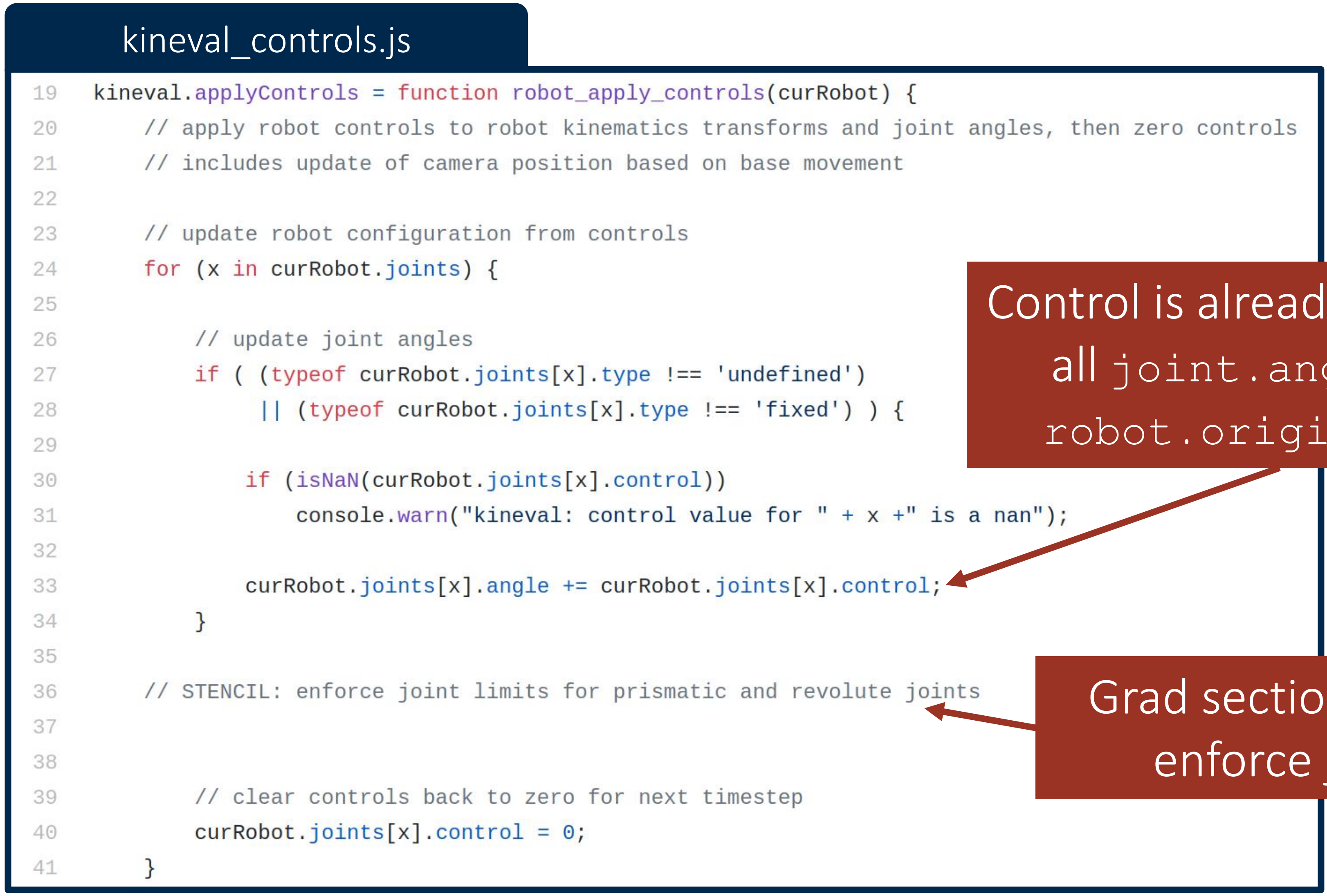
kineval_quaternion.js

```
///////////////////////////////////////////////////
/////     QUATERNION TRANSFORM ROUTINES
///////////////////////////////////////////////////

// STENCIL: reference quaternion code has the following functions:
//   quaternion_from_axisangle
//   quaternion_normalize
//   quaternion_to_rotation_matrix
//   quaternion_multiply

// **** Function stencils are provided below, please uncomment and implement them ****//

// kineval.quaternionFromAxisAngle = function quaternion_from_axisangle(axis,angle) {
```

Define quaternion helper functions
→ Create a joint's rotation matrix
from any axis-angle pair


Joint frame without control
Rest of forward kinematics
joint.angle
(dynamic w/ control)
joint.axis
(constant)
Joint frame with control
Rest of forward kinematics

# kineval_controls.js

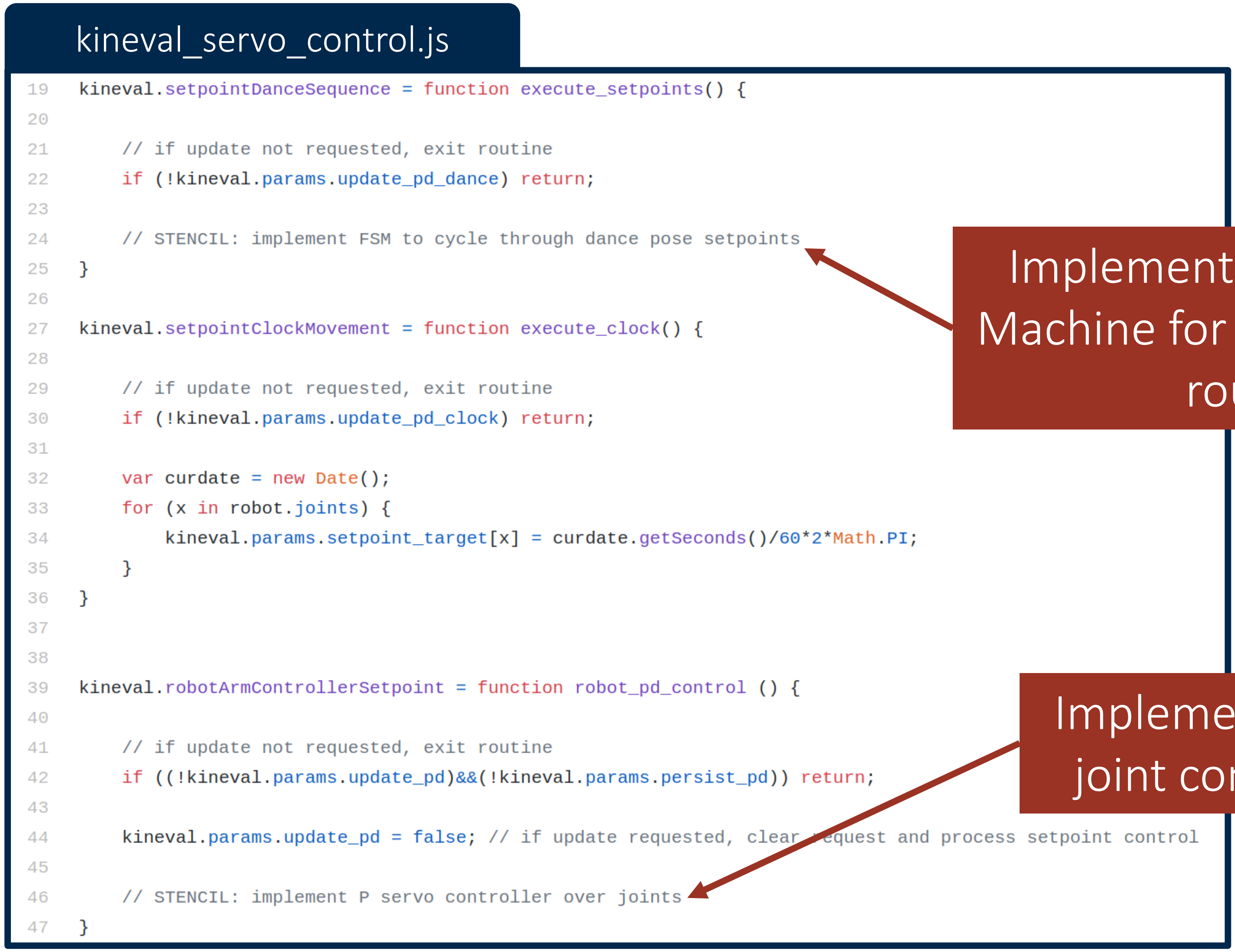

kineval_controls.js

```
kineval.applyControls = function robot_apply_controls(curRobot) {
    // apply robot controls to robot kinematics transforms and joint angles, then zero controls
    // includes update of camera position based on base movement

    // update robot configuration from controls
    for (x in curRobot.joints) {

        // update joint angles
        if ( (typeof curRobot.joints[x].type !== 'undefined')
             || (typeof curRobot.joints[x].type !== 'fixed') ) {

            if (isNaN(curRobot.joints[x].control))
                console.warn("kineval: control value for " + x +" is a nan");

            curRobot.joints[x].angle += curRobot.joints[x].control;
        }

    // STENCIL: enforce joint limits for prismatic and revolute joints


        // clear controls back to zero for next timestep
        curRobot.joints[x].control = 0;
    }
```

Control is already applied to all `joint.angle`s and `robot.origin` for you

Grad section will need to enforce joint limits

# kineval_servo_control.js



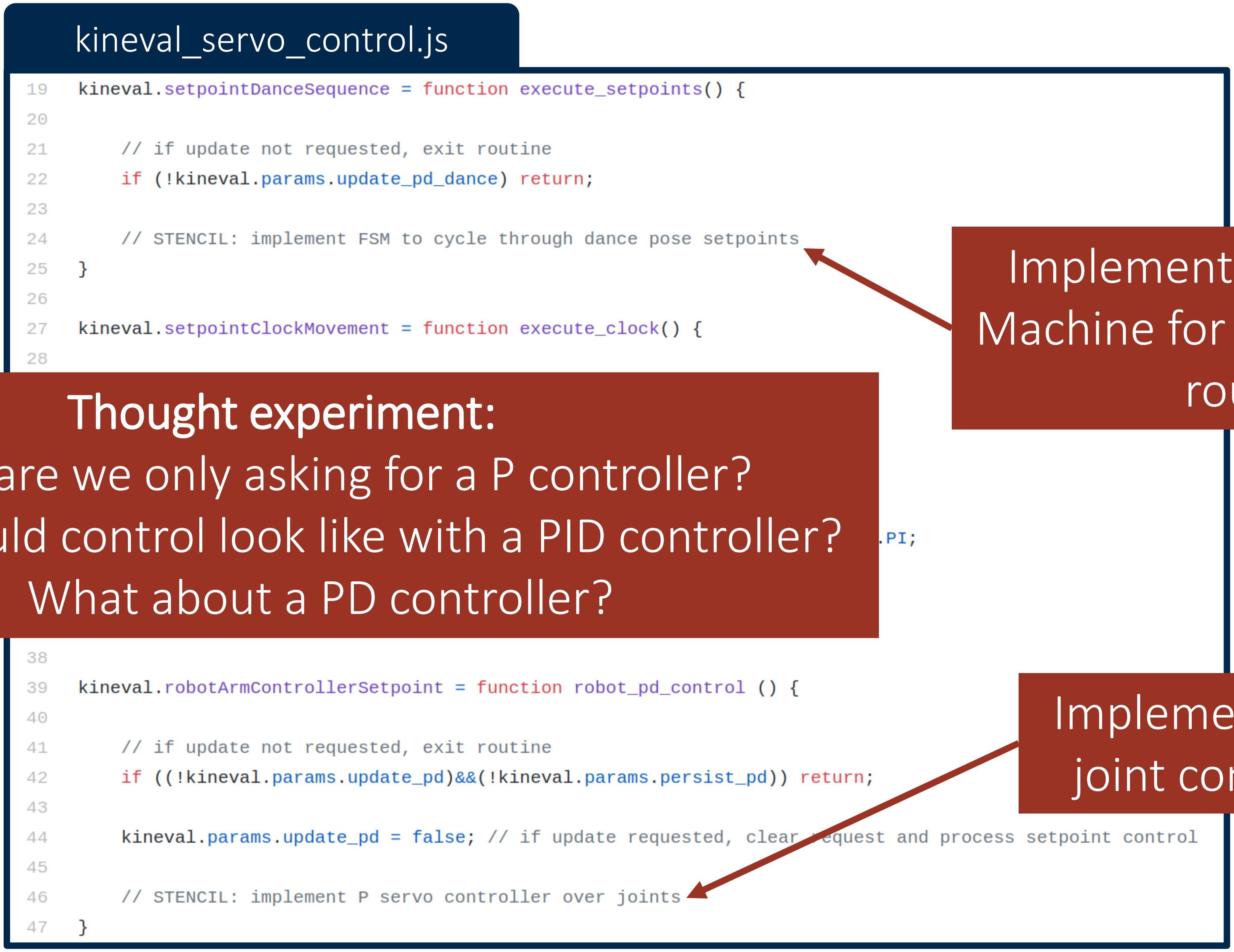
kineval_servo_control.js

```
kineval.setpointDanceSequence = function execute_setpoints() {

    // if update not requested, exit routine
    if (!kineval.params.update_pd_dance) return;

    // STENCIL: implement FSM to cycle through dance pose setpoints
}

kineval.setpointClockMovement = function execute_clock() {

    // if update not requested, exit routine
    if (!kineval.params.update_pd_clock) return;

    var curdate = new Date();
    for (x in robot.joints) {
        kineval.params.setpoint_target[x] = curdate.getSeconds()/60*2*Math.PI;
    }
}


kineval.robotArmControllerSetpoint = function robot_pd_control () {

    // if update not requested, exit routine
    if ((!kineval.params.update_pd)&&(!kineval.params.persist_pd)) return;

    kineval.params.update_pd = false; // if update requested, clear request and process setpoint control

    // STENCIL: implement P servo controller over joints
}
```

Implement a Finite State Machine for setpoint dance routine

Implement P controller for joint control to setpoints

# kineval_servo_control.js



```
kineval.setpointDanceSequence = function execute_setpoints() {

    // if update not requested, exit routine
    if (!kineval.params.update_pd_dance) return;

    // STENCIL: implement FSM to cycle through dance pose setpoints
}

kineval.setpointClockMovement = function execute_clock() {

.PI;

kineval.robotArmControllerSetpoint = function robot_pd_control () {

    // if update not requested, exit routine
    if ((!kineval.params.update_pd)&&(!kineval.params.persist_pd)) return;

    kineval.params.update_pd = false; // if update requested, clear request and process setpoint control

    // STENCIL: implement P servo controller over joints
}
```

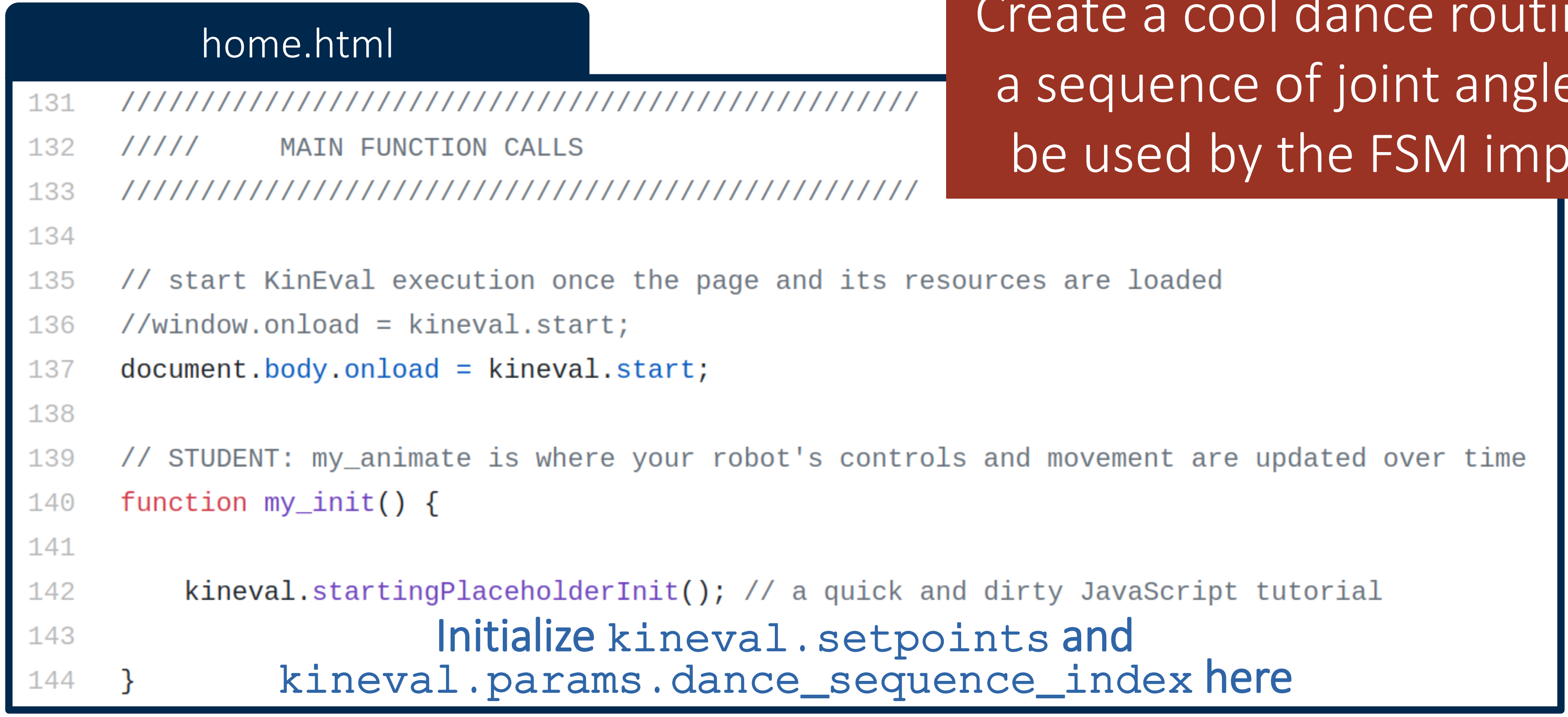
Implement a Finite State Machine for setpoint dance routine

**Thought experiment:**

1. Why are we only asking for a P controller?
2. What would control look like with a PID controller?
3. What about a PD controller?

Implement P controller for joint control to setpoints

# home.html



home.html

```
//////////////////////////////////////////////////
/////     MAIN FUNCTION CALLS
//////////////////////////////////////////////////

// start KinEval execution once the page and its resources are loaded
//window.onload = kineval.start;
document.body.onload = kineval.start;

// STUDENT: my_animate is where your robot's controls and movement are updated over time
function my_init() {

    kineval.startingPlaceholderInit(); // a quick and dirty JavaScript tutorial

}
```

Initialize `kineval.setpoints` and `kineval.params.dance_sequence_index` here

Create a cool dance routine by defining a sequence of joint angle setpoints to be used by the FSM implementation

Poses for servo can be set and stored **interactively** in KinEval using [0-9] keys and Shift+[0-9]

`JSON.stringify(kineval.setpoints)` will output the currently available servo setpoints to the console as a string

## Demo

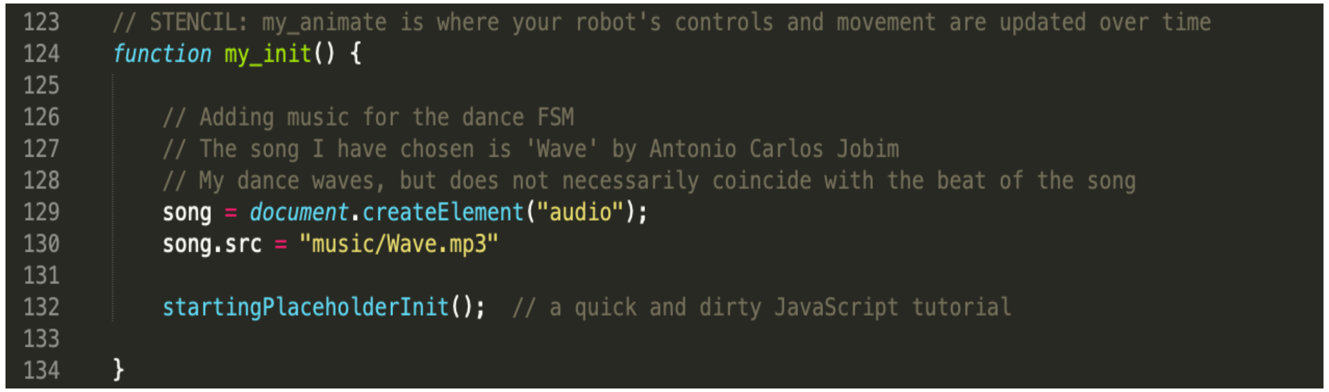
# HTML5 Audio

With two small additions to the stencil code, you can add music for your dance routine!

Uses the audio element offered by HTML5

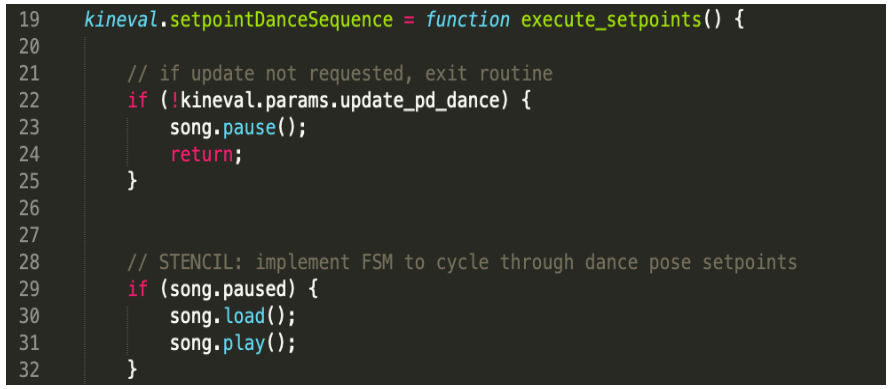
- We can load a song in home.html
- Then our FSM can play/pause the song along with the dance

**home.html**



```
// STENCIL: my_animate is where your robot's controls and movement are updated over time
function my_init() {

    // Adding music for the dance FSM
    // The song I have chosen is 'Wave' by Antonio Carlos Jobim
    // My dance waves, but does not necessarily coincide with the beat of the song
    song = document.createElement("audio");
    song.src = "music/Wave.mp3"

    startingPlaceholderInit();  // a quick and dirty JavaScript tutorial

}
```

**kineval_servo_control.js**



```
kineval.setpointDanceSequence = function execute_setpoints() {

    // if update not requested, exit routine
    if (!kineval.params.update_pd_dance) {
        song.pause();
        return;
    }


    // STENCIL: implement FSM to cycle through dance pose setpoints
    if (song.paused) {
        song.load();
        song.play();
    }
```