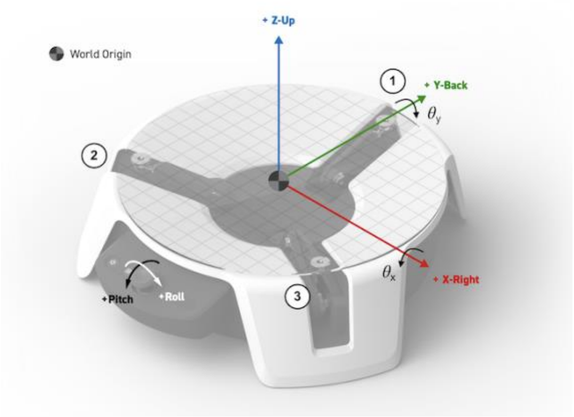
Moab Ball Balance Sample Bonsai Brain- AI Solution Spec

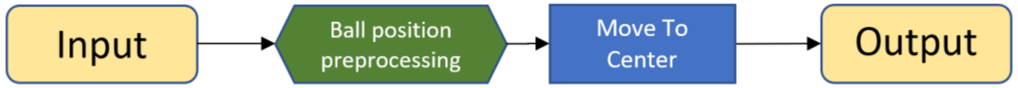
Project start/end dates**:** Click here to enter a date.

Authors: John Alexander

Customer	**Moab Device Ball Balance**	
Project Objective	*What task/process are you looking to improve using deep reinforcement learning?*	• The device (shown in Figure 1) has three arms powered by servo motors and runs off a Raspberry Pi 4. • These arms work in tandem to control the angle of the transparent plate to keep the ball balanced. • The Moab device tracks and maps the ball movement onto a standard 2D coordinate system (shown in Figure 2) using computer vision. • Looking at the front the of the device, the x-axis runs left-to-right, and the y-axis runs front-to-back, with the plate center at location (0, 0), and a radius of r. • The trained AI must learn how to adjust the angle of the plate to balance a ball.  **Figure 1**: Moab Device  **Figure 2**: Diagram of the Z-Up right hand coordinate system used by the Moab device.
Business Value	*What is the business value of improving the*	• Improved balancing reduces energy consumption and wear on motors

<table>
<tr><td></td><td>control/optimization of this system?</td><td>Reduced time to balance leads to more efficient demos</td></tr>
<tr><td>Optimization Goal</td><td>What Key Performance Indicators (KPI) define the control or optimization of this system?</td><td>
<table>
<tr><th>Goal (KPI)</th><th>Units</th><th>Description</th></tr>
<tr><td>Settling time</td><td>seconds</td><td>Amount of time it takes the ball to reach the center of the plate and stay there</td></tr>
</table>
</td></tr>
<tr><td>Current Methods</td><td>How do you currently control or optimize the system?</td><td>
<table>
<tr><th></th><th>Method</th><th>Level</th></tr>
<tr><td>☐</td><td>Human Operator</td><td></td></tr>
<tr><td>☐</td><td>Expert System</td><td></td></tr>
<tr><td>x☒</td><td>Control Theory (PID, MPC)</td><td>Low-Level Control: PID</td></tr>
<tr><td>☐</td><td>Advanced Process Control (APC)</td><td></td></tr>
<tr><td>☐</td><td>Optimization Techniques</td><td></td></tr>
</table>
</td></tr>
</table>

<table>
<tr>
<td>Limitations of Current Methods</td>
<td>What are the challenges and limitations of the current method(s)?</td>
<td>
<table>
<tr><th></th><th>Limitation</th><th>Description</th></tr>
<tr><td>☒</td><td>Ability to control well across scenarios / conditions</td><td></td></tr>
<tr><td>☐</td><td>Multiple or changing optimization goals</td><td></td></tr>
<tr><td>☐</td><td>Human Operator / Engineer Limitations</td><td>
<table>
<tr><th></th><th>Limitation</th><th>Details</th></tr>
<tr><td>☐</td><td>Difficulty managing many variables and dimensions.</td><td></td></tr>
<tr><td>☐</td><td>Difficulty adapting to changing conditions</td><td></td></tr>
<tr><td>☐</td><td>Large performance discrepancy between novice and expert operators</td><td>.</td></tr>
<tr><td>☐</td><td>Inconsistency across expert operators</td><td></td></tr>
</table>
</td></tr>
<tr><td>☒</td><td>Uncertainty in the measurement of the inputs or the process make it difficult to control or optimize.</td><td></td></tr>
<tr><td>☐</td><td>Time to develop control or optimization system is prohibitive</td><td></td></tr>
</table>
</td>
</tr>
<tr>
<td>Machine Teaching Strategy</td>
<td colspan="2">Concept Decomposition - Monolithic</td>
</tr>
</table>



- Ball Position tracking provided by image preprocessor

Concept 1: Move To Center

- The trained AI must learn how to adjust the plate pitch and roll to balance a ball using the following objectives (as referenced in Figure 2):
    - The ball position (x, y) will reach the plate center at (0, 0) and stay there.
    - The ball position will not get near the plate edge at (| (x, y) - (0, 0) | << r).

Control Actions

What actions will the brain need to output to control or optimize your system?

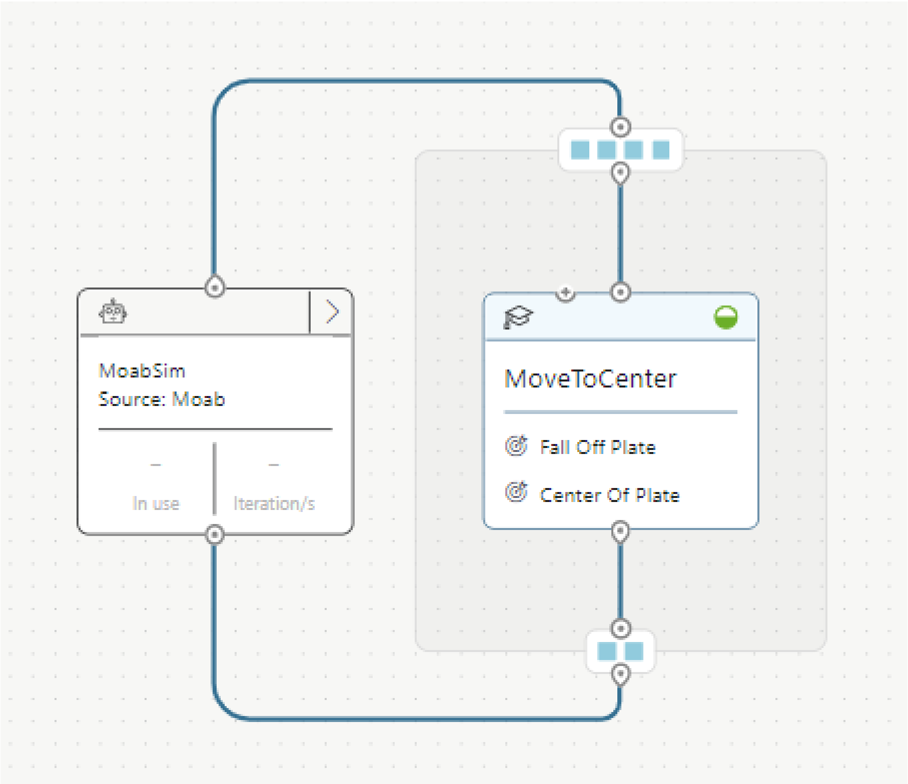
	Level	Number of Actions	Description
☐	Supervisory		The brain will provide supervisory set points.
x ☒	Low-level	2	Low-level control will remain with the servo controllers.

Name	Data Type	Units	Control Frequency	Operating Range [min, max]	Description
input_pitch	Decimal	radians	30 Hz: 0.0333333333333333 s(p)	[-1,1]	Change in pitch of the full plate rotation range supported by the hardware.at each timestep
input_roll	Decimal	radians	30 Hz: 0.0333333333333333 s(p)	[-1,1]	Change in roll of the full plate rotation range supported by the hardware.at each timestep

<table>
<tr>
<td>Constraints</td>
<td><i>What constraints are placed on the control actions by the system or the process?</i></td>
<td></td>
</tr>
<tr>
<td>Environment States</td>
<td><i>What information do we need to pass to the brain about the system and its environment for the brain to learn to control or optimize the system?</i></td>
<td>
<table>
<tr><th>Name</th><th>Data Type</th><th>Source</th><th>Units</th><th>Measurement Frequency</th><th>Operating Range [min, max]</th><th>Description</th></tr>
<tr><td>ball_x</td><td>Decimal</td><td>Simulation</td><td>meters</td><td>[frequency]</td><td>[-0.3825, 0.3825]</td><td>position of ball from center of plate along x-axis</td></tr>
<tr><td>ball_y</td><td>Decimal</td><td>Simulation</td><td>meters</td><td>[frequency]</td><td>[-0.3825, 0.3825]</td><td>position of ball from center of plate along y-axis</td></tr>
<tr><td>ball_vel_x</td><td>Decimal</td><td>Simulation</td><td>meters/sec</td><td>[frequency]</td><td>[-6.0, 6.0]</td><td>X-velocity of ball</td></tr>
<tr><td>ball_vel_y</td><td>Decimal</td><td>Simulation</td><td>meters/sec</td><td>[frequency]</td><td>[-6.0, 6.0]</td><td>Y- velocity of ball</td></tr>
</table>
</td>
</tr>
</table>

Deep Reinforcement Learning

Deep Reinforcement Learning algorithms train agents to make sequential decisions which are assessed for the effect that each decision has on the environment.



For each concept that we will train using Deep Reinforcement Learning, we outline the sequential decision.

Concept	**Action**	**State**: How does the Environment change when the control actions are taken?	**Reward**	**Configuration**: What do we need to vary in the training to ensure that the brain works well across scenarios?
Move To Center	Adjust plate pitch and roll	Each time a decision is made the ball position and velocity changes	• Avoid Fall Off Plate - Ball's distance from the center must not reach values above 80% of the plate radius. • Drive Center Of Plate - Ball's X and Y coordinates (state.ball_x and state.ball_y) must stay within CloseEnough radial distance from the plate center.	• Initial ball conditions (initial_x, initial_y) • Initial ball velocity conditions (initial_vel_x, initial_vel_y) • Initial plate conditions (initial_pitch, initial_roll) • Optional for Robustness: Ball properties (shell and radius)

<table>
<tr>
<td>Configuration Scenarios</td>
<td><i>What scenarios should the trained brain be able to control across?</i></td>
<td>
Deep Reinforcement Learning (DRL) can produce brain(s) that control well across a wide range of scenarios and is particularly suitable for situations where the distribution of the variables in the configuration scenarios is unknown and / or non-linear.
<table>
<tr><th>Configuration Variable</th><th>Range [min, max]</th><th>Description</th></tr>
<tr><td>initial_x:</td><td>[-0.05625, 0.05625]</td><td>Ball X position</td></tr>
<tr><td>initial_y:</td><td>[-0.05625, 0.05625]</td><td>Ball Y position</td></tr>
<tr><td>initial_vel_x:</td><td>[-0.02, 0.02]</td><td>Ball X velocity</td></tr>
<tr><td>initial_vel_y:</td><td>[-0.02, 0.02]</td><td>Ball Y velocity</td></tr>
<tr><td>initial_pitch</td><td>[-0.2, 0.2]</td><td>Plate pitch</td></tr>
<tr><td>initial_roll</td><td>[-0.2, 0.2]</td><td>Plate roll</td></tr>
</table>
Training Episode Length: 250 control actions
Benchmark Episode Length: 180 control actions
</td>
</tr>
<tr>
<td>Success Criteria</td>
<td><i>What criteria will we use to determine the success of the project and how will we measure that success criteria?</i></td>
<td>
<table>
<tr><td>KPI</td><td>Ball in center of plate</td></tr>
<tr><td>Benchmark Comparison</td><td>The brain will be compared to a PID controller</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td>
<table>
<tr>
<td>Benchmark Scenarios</td>
<td>
<table>
<tr><th>Configuration Variable</th><th>Units</th><th>Priority</th><th>Range or Description</th></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>
</td>
</tr>
<tr>
<td>Benchmark Procedure</td>
<td>
<table>
<tr><th></th><th>Procedure</th><th>Duration</th></tr>
<tr><td>☒</td><td>Simulation</td><td>[Benchmark Duration in Simulation]</td></tr>
<tr><td>x☐</td><td>A/B Testing on Live System</td><td>[A/B Testing on Live System]</td></tr>
</table>
</td>
</tr>
<tr>
<td>Optimization Improvement</td>
<td>[success criteria expressed in % improvement over current methods]</td>
</tr>
<tr>
<td>Return on Investment (ROI)</td>
<td>[success criteria expressed in return on investment (ROI)]</td>
</tr>
<tr>
<td>Project Readout and Deliverables</td>
<td>[expected deliverables besides the brain(s) and a PowerPoint readout report]</td>
</tr>
</table>
<p>☒ Is the benchmark controller or performance noted above currently ready?</p>
</td>
</tr>
<tr>
<td>Simulation</td>
<td></td>
<td>
<table>
<tr>
<td>Readiness</td>
<td>
<table>
<tr><td>Delivery Date</td><td>[Sim Delivery Date]</td></tr>
<tr><td>Validation Date</td><td>[Sim Validation Date]</td></tr>
<tr><td>Sim Builder</td><td>[Sim Builder]</td></tr>
<tr><td>Integration with Microsoft Machine Teaching Service</td><td>SDK3</td></tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>

Type

Vendor	
Product (Version)	N/A, custom sim
Software Language API Interface	Python
Speed	Less than 1 second per iteration

Modeling Method

	Method	Description
☒	Physics Based (First Principles)	Python simulator of Moab
☐	Discrete Event	
☐	Surrogate Model	
☐	Model from Data	The amount (number of rows) of data required to create a simulation model from data varies, but use the following rule of thumb as an absolute minimum: the number of possible states x the number of possible actions. For example, if there are 10 possible actions and 100 possible states, you'd need 1,000 rows of data at minimum to build a model.

Model Accuracy & Robustness

The model should be validated across the ranges for each of the control actions and environment states listed above. Enter the accuracy of the model for each of the features.

Feature	Accuracy
[One row for each control action and environment state]	[% Error]

State Space Completion

	Procedure	Rows of Data
☐	State Space Parameter Sweep	[data volume]
☐	Synthetic State Space Estimation	[data volume]

Connection

Can we exchange messages (input and output) with the simulation model at the simulated control frequency?

Is a high-level control system diagram from sensors to actuators available?

	Are there any other pieces required, beside the simulator, to run the training loop? [Can this software connect for input and output on the inner loop?]
Configuration	*Can we input the configuration scenarios programmatically into the simulation model?* [Can we input configuration scenarios programmatically into the simulation model?]
Parallelization (Licensing)	*Can we run 10, 100 or 1000 copies of your simulation in the Azure cloud?* [Can we run the simulation in the Azure Cloud?]
Simulation to Reality	*Has the simulation model been used to design a control system, an optimization system or used by human operators in production to control the system.* No. *What is the error percentage that describes the accuracy between the simulation model and the real system across all scenarios and equipment that will be controlled by the brain?* [% error] *Do you plan any simulator upgrades, especially if it will need to be upgraded for use with Bonsai?*
Simulation Validation	*Are there any major assumptions in the sim that would change the sim dynamics as compared to the real-world dynamics?* *Can you provide the validation data against your sim?* *Are there special, external libraries?* *Does the model connect to external data sources?* *How many workarounds are needed to setup the model to run headlessly, ideally using only parameters on the top level agent (Main)?*

<table>
<tr>
<td>Supplementary Decision Models</td>
<td>Will Machine Learning (ML) models or other decision-making technology be used to supplement the environment state from the simulator?</td>
<td>
<table>
<tr><th>Type</th><th>Training Data</th><th>Model Accuracy</th><th>Description</th></tr>
<tr><td>Liner Predictor</td><td>[training data]</td><td>[model accuracy]</td><td>[description]</td></tr>
<tr><td>ROM Bin Input Predictor</td><td>[training data]</td><td>[model accuracy]</td><td>[description]</td></tr>
<tr><td>COB Bin Output Predictor</td><td>[training data]</td><td>[model accuracy]</td><td>[description]</td></tr>
<tr><td>Rock Hardness Fragmentation Predictor</td><td>[training data]</td><td>[model accuracy]</td><td>[description]</td></tr>
<tr><td>Maximum Safe Crusher Gap Predictor</td><td>[training data]</td><td>[model accuracy]</td><td>[description]</td></tr>
</table>
<table>
<tr><td>Delivery Date</td><td>[Delivery Date]</td></tr>
<tr><td>Validation Date</td><td>[Validation Date]</td></tr>
<tr><td>Model Builder</td><td></td></tr>
<tr><td>Integration with Microsoft Machine Teaching Service</td><td>Python SDK2</td></tr>
</table>
</td>
</tr>
</table>

Deployment

How will the brain interface with your system? (select & respond to one or multiple options below)

Is the deployment interface and protocol defined and ready?

If it does not exist, what is the delivery date?

☐ **Decision Support**

Human engineers, operators or analysts will continue to control and automate my system augmented by brain decisions.

☐ **Cloud Deployment**

☐ **Edge Deployment**

☐ **Embedded Deployment**

Integration with OT environment will be through existing IoT/OPC Gateway
Edge infrastructure enabled to host containers, within the IT environment.

	Decision Delivery Mechanism	Description
☒	Decision Support UI	
☐	Spreadsheet or other mechanism	
☐	Integration with current reporting system	

☐ **Direct Control**

The brain will connect to the system directly to automate the control or optimization.

☐ **Cloud Deployment**

☐ **Edge Deployment**

☐ **Embedded Deployment**

For Edge and Embedded Deployments:

Device Type	[Device Type]
Number of Devices	[Number of Devices]
Device Lifecycle	[Device Lifecycle]
Docker Support	[Docker Support]
Processor	[Processor]
Connection Protocol	[Connection Protocol]
Integrator	[Integrator]
Integration Delivery Date	[Integration Delivery Date]

Team		**Executive Sponsor**	[Executive Sponsor Name]
		Machine Teacher	[Machine Teacher Name]
		Data Scientist (Optional)	[Data Scientist Name]
		Subject Matter Expert	[Subject Matter Expert Name]
		Simulation Expert	[Simulation Expert Name]
		Deployment Expert	[Deployment Expert Team]
		IT	[IT Contact Name]
		Project Team	[]
		Services Partner	[Services Partner Team]
		Microsoft Applied AI Engineer	[Project Applied AI Engineer]
		Microsoft Technical Program Manager	[Project Technical Program Manager]
		Microsoft Account Team	[Account Executive, Account Technical Strategist Names]
		Microsoft CSA	[CSA Name]
Azure Infrastructure		**Azure Subscription**	[Subscription ID]
		Other Azure Services Required	[List of Azure Services]

		Will Microsoft be given access to the customer’s Azure subscription?