



دانشگاه اصفهان  
دانشکده مهندسی کامپیوتر

هوش محاسباتی  
گزارش کتبی تمرین دوم  
بخش سوم: طراحی سیستم منطق فازی برای توصیه تمرینات ورزشی

اعضای گروه:

رضا چراخ

فرناز موحدی

اردیبهشت ۱۴۰۴

## فهرست مطالب

صفحه

عنوان

2	فصل اول پیاده‌سازی پروژه و تحلیل نتایج .....
2-۱-۱	مقدمه .....
2-۲-۱	تعریف متغیرهای ورودی فازی .....
3-۳-۱	تعریف متغیرهای خروجی فازی .....
3-۴-۱	تعریف توابع عضویت فازی برای هر متغیر .....
4-۱-۴-۱	متغیرهای ورودی فازی .....
4-۲-۴-۱	متغیرهای خروجی فازی .....
5-۳-۴-۱	نمودارهای توابع عضویت .....
6-۵-۱	تعریف قواعد فازی .....
6-۱-۵-۱	قوانین مربوط به شدت تمرین .....
7-۲-۵-۱	قوانین مربوط به مدت زمان تمرین .....
9-۳-۵-۱	بازتعریف برخی قوانین براساس ترجیحات کاربر .....
9-۶-۱	استفاده از استنتاج فازی .....
10-۷-۱	تست و اعتبارسنجی سیستم .....
10-۱-۷-۱	تابع test_system() .....
11-۲-۷-۱	تابع کمکی categorize_output .....
12-۳-۷-۱	تست کردن تابع test_system() .....
13	منابع .....

## فصل اول

### پیاده‌سازی پروژه و تحلیل نتایج

#### ۱-۱- مقدمه

در این پروژه، یک سیستم فازی طراحی شده که بر اساس پنج ورودی کلیدی آمادگی جسمانی، میزان انرژی روزانه، سن، وزن و هدف تمرینی (مثلاً کاهش وزن یا افزایش عضله)، شدت (Intensity) و مدت زمان (Duration) مناسب تمرین را توصیه می‌کند.

#### ۱-۲- تعریف متغیرهای ورودی فازی

در این بخش، پنج متغیر ورودی تعریف شده‌اند:

- Fitness: سطح آمادگی جسمانی از ۰ (مبتدی) تا ۱۰ (پیشرفته).
- Energy: سطح انرژی از ۰ (کم) تا ۱۰ (زیاد).
- Goal: هدف ورزشی شامل ۳ دسته: کاهش وزن (۰)، افزایش عضله (۱)، تناسب عمومی (۲).
- Age: سن بین ۱۵ تا ۸۰ سال.
- Weight: وزن بین ۴۰ تا ۱۲۰ کیلوگرم.

```
# Define fuzzy variables
fitness = ctrl.Antecedent(np.arange(0, 11, 1), 'fitness')
energy = ctrl.Antecedent(np.arange(0, 11, 1), 'energy')
goal = ctrl.Antecedent(np.arange(0, 3, 1), 'goal')
age = ctrl.Antecedent(np.arange(15, 81, 1), 'age')
weight = ctrl.Antecedent(np.arange(40, 121, 1), 'weight')
```

### ۳-۱- تعریف متغیرهای خروجی فازی

در این بخش، دو متغیر خروجی تعریف شده‌اند:

- **Intensity**: شدت تمرین بین ۰ تا ۱۰.
- **Duration**: مدت تمرین بین ۰ تا ۹۰ دقیقه.

```
intensity = ctrl.Consequent(np.arange(0, 11, 1), 'intensity')
duration = ctrl.Consequent(np.arange(0, 91, 1), 'duration')
```

### ۴-۱- تعریف توابع عضویت فازی برای هر متغیر

برای هر متغیر، مجموعه‌های فازی تعریف شده‌اند تا مقادیر عددی را به دسته‌های زبانی تبدیل کنند. انتخاب نوع تابع عضویت بستگی به ویژگی‌های داده‌ها و نیاز سیستم دارد. در این سیستم فازی، از دو نوع تابع عضویت استفاده شده است:

- توابع مثلثی: (Trimf)
    - این توابع برای دسته‌هایی استفاده می‌شوند که دارای یک نقطه مرکزی واضح هستند و تغییرات سریع دارند.
    - به‌عنوان مثال، هدف‌های تمرینی مانند "کاهش وزن" یا "افزایش عضله" به‌طور طبیعی دارای یک نقطه بهینه هستند و تغییرات تدریجی در این بازه‌ها می‌تواند معنادار باشد.
  - توابع دوزنقه‌ای: (Trapmf)
    - این توابع برای دسته‌هایی استفاده می‌شوند که به نوعی در محدوده‌های بزرگتر ثابت هستند و تغییرات تدریجی‌تری دارند.
    - به‌عنوان مثال، سطح آمادگی جسمانی ممکن است در ابتدا یا انتهای بازه فازی ثابت باشد (برای "مبتدی" یا "پیشرفته")، و تنها در میانه‌های بازه تغییرات معناداری داشته باشد.
- استفاده از این دو نوع تابع عضویت باعث می‌شود که مدل قادر به تعیین شدت و مدت تمرین بر اساس ورودی‌های مختلف به‌طور دقیق و معقول باشد.

#### ۱-۴-۱- متغیرهای ورودی فازی

```
# Fitness
fitness['beginner'] = fuzz.trapmf(fitness.universe, [0, 0, 2, 4])
fitness['intermediate'] = fuzz.trimf(fitness.universe, [3, 5, 7])
fitness['advanced'] = fuzz.trapmf(fitness.universe, [6, 8, 10, 10])

# Energy
energy['low'] = fuzz.trapmf(energy.universe, [0, 0, 2, 4])
energy['medium'] = fuzz.trimf(energy.universe, [3, 5, 7])
energy['high'] = fuzz.trapmf(energy.universe, [6, 8, 10, 10])

# Goal
goal['weight_loss'] = fuzz.trimf(goal.universe, [-0.5, 0, 0.5])
goal['muscle_gain'] = fuzz.trimf(goal.universe, [0.5, 1, 1.5])
goal['general_fitness'] = fuzz.trimf(goal.universe, [1.5, 2, 2.5])

# Age
age['young'] = fuzz.trapmf(age.universe, [15, 15, 20, 35])
age['middle'] = fuzz.trimf(age.universe, [30, 45, 60])
age['elderly'] = fuzz.trapmf(age.universe, [55, 70, 80, 80])

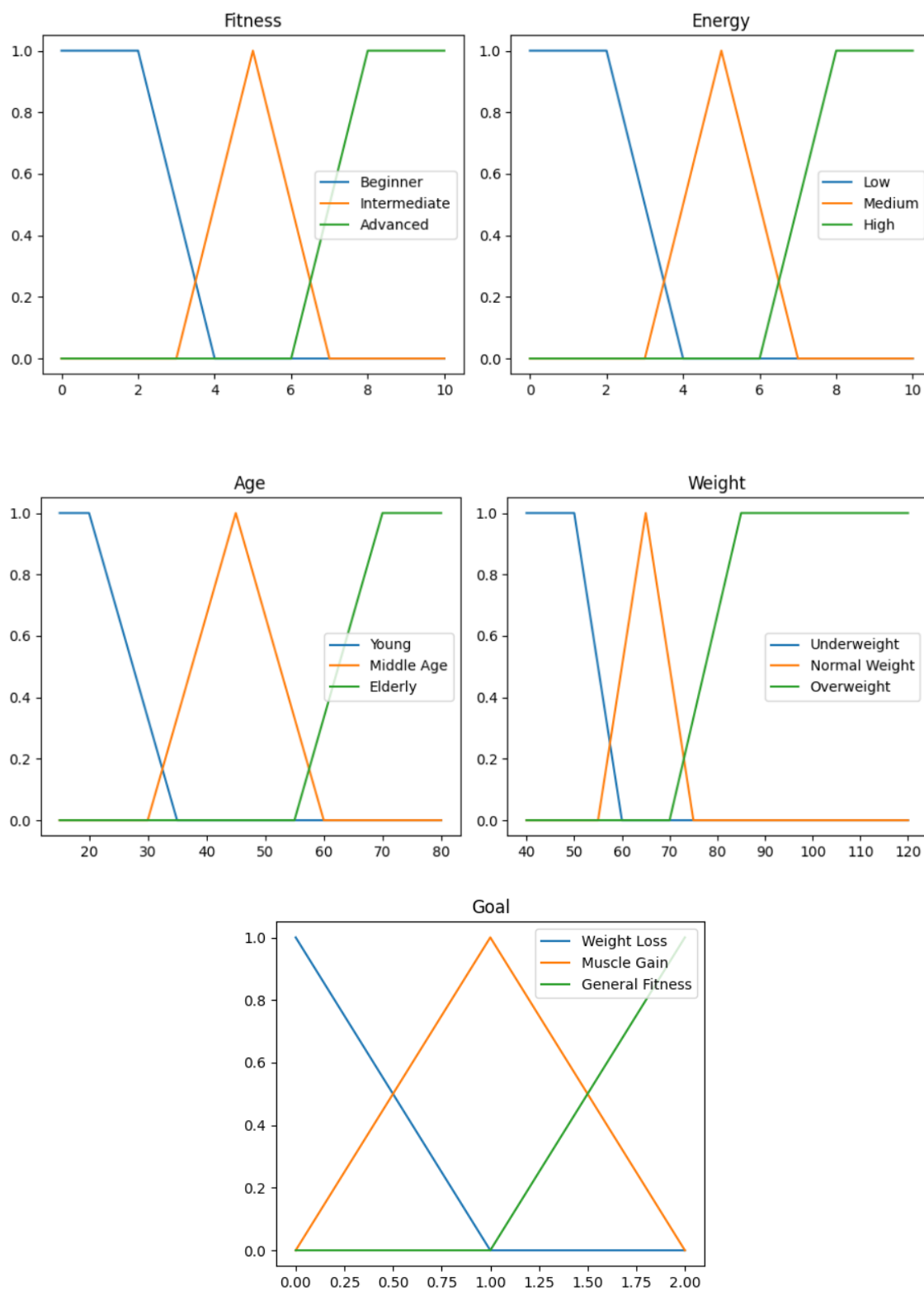
# Weight
weight['under'] = fuzz.trapmf(weight.universe, [40, 40, 50, 60])
weight['normal'] = fuzz.trimf(weight.universe, [55, 65, 75])
weight['over'] = fuzz.trapmf(weight.universe, [70, 85, 120, 120])
```

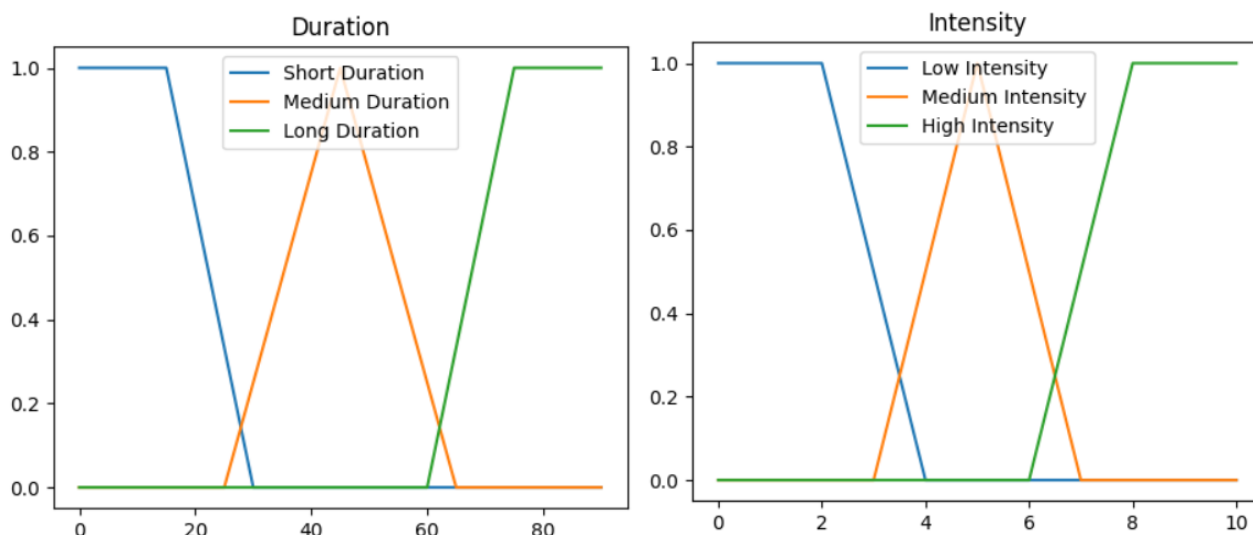
#### ۱-۴-۲- متغیرهای خروجی فازی

```
# Intensity
intensity['low'] = fuzz.trapmf(intensity.universe, [0, 0, 2, 4])
intensity['medium'] = fuzz.trimf(intensity.universe, [3, 5, 7])
intensity['high'] = fuzz.trapmf(intensity.universe, [6, 8, 10, 10])

# Duration
duration['short'] = fuzz.trapmf(duration.universe, [0, 0, 15, 30])
duration['medium'] = fuzz.trimf(duration.universe, [25, 45, 65])
duration['long'] = fuzz.trapmf(duration.universe, [60, 75, 90, 90])
```

### ۱-۴-۳- نمودارهای توابع عضویت





## ۵-۱- تعریف قواعد فازی

### ۵-۱-۱- قوانین مربوط به شدت تمرین

- اگر آمادگی جسمانی مبتدی باشد و انرژی کم باشد، شدت تمرین کم خواهد بود.
- اگر آمادگی جسمانی مبتدی باشد و انرژی متوسط باشد، شدت تمرین کم خواهد بود.
- اگر آمادگی جسمانی مبتدی باشد و انرژی بالا باشد، شدت تمرین متوسط خواهد بود.
- اگر آمادگی جسمانی متوسط باشد و انرژی کم باشد، شدت تمرین کم خواهد بود.
- اگر آمادگی جسمانی متوسط باشد و انرژی متوسط باشد، شدت تمرین متوسط خواهد بود.
- اگر آمادگی جسمانی متوسط باشد و انرژی بالا باشد، شدت تمرین بالا خواهد بود.
- اگر آمادگی جسمانی پیشرفته باشد و انرژی کم باشد، شدت تمرین متوسط خواهد بود.
- اگر آمادگی جسمانی پیشرفته باشد و انرژی متوسط باشد، شدت تمرین بالا خواهد بود.
- اگر آمادگی جسمانی پیشرفته باشد و انرژی بالا باشد، شدت تمرین بالا خواهد بود.
- اگر سن بالا باشد (سالمند)، شدت تمرین کم خواهد بود.
- اگر سن میانسال باشد و انرژی بالا باشد، شدت تمرین متوسط خواهد بود.
- اگر هدف افزایش عضله باشد و انرژی بالا باشد، شدت تمرین بالا خواهد بود.
- اگر هدف کاهش وزن باشد و آمادگی جسمانی مبتدی باشد، شدت تمرین متوسط خواهد بود.
- اگر وزن کمتر از حد نرمال باشد و هدف افزایش عضله باشد، شدت تمرین متوسط خواهد بود.
- اگر انرژی کم باشد و سن سالمند باشد، شدت تمرین کم خواهد بود.
- اگر هدف تناسب اندام عمومی باشد و آمادگی جسمانی مبتدی باشد، شدت تمرین کم خواهد بود.
- اگر هدف تناسب اندام عمومی باشد و آمادگی جسمانی متوسط باشد، شدت تمرین متوسط خواهد بود.
- اگر هدف تناسب اندام عمومی باشد و آمادگی جسمانی پیشرفته باشد، شدت تمرین بالا خواهد بود.
- اگر هدف افزایش عضله باشد و آمادگی جسمانی متوسط و انرژی متوسط باشد، شدت تمرین بالا خواهد بود.

بود.

اگر هدف کاهش وزن باشد و آمادگی جسمانی پیشرفته و انرژی متوسط باشد، شدت تمرین متوسط خواهد

بود.

اگر وزن کمتر از حد نرمال باشد و انرژی پایین باشد، شدت تمرین کم خواهد بود.

اگر وزن بیشتر از حد نرمال باشد و انرژی بالا باشد و هدف افزایش عضله باشد، شدت تمرین بالا خواهد

بود.

اگر سن میانسال باشد و انرژی پایین باشد، شدت تمرین کم خواهد بود.

اگر فرد جوان باشد و انرژی بالا باشد، شدت تمرین بالا خواهد بود.

```
# --- INTENSITY RULES ---
rule1 = ctrl.Rule(fitness['beginner'] & energy['low'], intensity['low'])
rule2 = ctrl.Rule(fitness['beginner'] & energy['medium'], intensity['low'])
rule3 = ctrl.Rule(fitness['beginner'] & energy['high'], intensity['medium'])
rule4 = ctrl.Rule(fitness['intermediate'] & energy['low'], intensity['low'])
rule5 = ctrl.Rule(fitness['intermediate'] & energy['medium'], intensity['medium'])
rule6 = ctrl.Rule(fitness['intermediate'] & energy['high'], intensity['high'])
rule7 = ctrl.Rule(fitness['advanced'] & energy['low'], intensity['medium'])
rule8 = ctrl.Rule(fitness['advanced'] & energy['medium'], intensity['high'])
rule9 = ctrl.Rule(fitness['advanced'] & energy['high'], intensity['high'])
rule10 = ctrl.Rule(age['elderly'], intensity['low'])
rule11 = ctrl.Rule(age['middle'] & energy['high'], intensity['medium'])
rule12 = ctrl.Rule(goal['muscle_gain'] & energy['high'], intensity['high'])
rule13 = ctrl.Rule(goal['weight_loss'] & fitness['beginner'], intensity['medium'])
rule14 = ctrl.Rule(weight['under'] & goal['muscle_gain'], intensity['medium'])
rule15 = ctrl.Rule(energy['low'] & age['elderly'], intensity['low'])
rule16 = ctrl.Rule(goal['general_fitness'] & fitness['beginner'], intensity['low'])
rule17 = ctrl.Rule(goal['general_fitness'] & fitness['intermediate'], intensity['medium'])
rule18 = ctrl.Rule(goal['general_fitness'] & fitness['advanced'], intensity['high'])
rule19 = ctrl.Rule(goal['muscle_gain'] & fitness['intermediate'] & energy['medium'], intensity['high'])
rule20 = ctrl.Rule(goal['weight_loss'] & fitness['advanced'] & energy['medium'], intensity['medium'])
rule21 = ctrl.Rule(weight['under'] & energy['low'], intensity['low'])
rule22 = ctrl.Rule(weight['over'] & energy['high'] & goal['muscle_gain'], intensity['high'])
rule23 = ctrl.Rule(age['middle'] & energy['low'], intensity['low'])
rule24 = ctrl.Rule(age['young'] & energy['high'], intensity['high'])
```

## ۱-۵-۲- قوانین مربوط به مدت زمان تمرین

اگر هدف کاهش وزن باشد و وزن بیشتر از حد نرمال باشد، مدت تمرین طولانی خواهد بود.

اگر هدف کاهش وزن باشد و انرژی پایین باشد، مدت تمرین متوسط خواهد بود.

اگر هدف افزایش عضله باشد، مدت تمرین متوسط خواهد بود.

اگر هدف تناسب اندام عمومی باشد و انرژی پایین باشد، مدت تمرین کوتاه خواهد بود.

اگر هدف تناسب اندام عمومی باشد و انرژی بالا باشد، مدت تمرین متوسط خواهد بود.

اگر آمادگی جسمانی پیشرفته باشد و انرژی بالا باشد، مدت تمرین طولانی خواهد بود.

اگر آمادگی جسمانی مبتدی باشد و انرژی کم باشد، مدت تمرین کوتاه خواهد بود.

اگر سن بالا باشد (سالمند)، مدت تمرین کوتاه خواهد بود.



اگر سن میانسال باشد و انرژی متوسط باشد، مدت تمرین متوسط خواهد بود.

اگر فرد جوان باشد و انرژی بالا باشد، مدت تمرین طولانی خواهد بود.

اگر وزن کمتر از حد نرمال باشد و هدف افزایش عضله باشد، مدت تمرین متوسط خواهد بود.

اگر وزن نرمال باشد و هدف تناسب اندام عمومی باشد، مدت تمرین متوسط خواهد بود.

اگر وزن بیشتر از حد نرمال باشد و انرژی پایین باشد، مدت تمرین کوتاه خواهد بود.

اگر آمادگی جسمانی متوسط باشد و هدف افزایش عضله باشد، مدت تمرین متوسط خواهد بود.

اگر هدف کاهش وزن باشد و آمادگی جسمانی پیشرفته باشد، مدت تمرین طولانی خواهد بود.

اگر هدف تناسب اندام عمومی باشد و انرژی متوسط باشد، مدت تمرین متوسط خواهد بود.

اگر سن بالا باشد و انرژی بالا باشد، با وجود انرژی زیاد، مدت تمرین فقط متوسط خواهد بود.

اگر آمادگی جسمانی پیشرفته باشد و هدف تناسب اندام عمومی باشد، مدت تمرین متوسط خواهد بود.

اگر وزن کمتر از حد نرمال باشد و هدف کاهش وزن باشد، مدت تمرین کوتاه خواهد بود.

اگر فرد جوان باشد و انرژی متوسط داشته باشد، مدت تمرین متوسط خواهد بود.

اگر آمادگی جسمانی متوسط باشد و هدف تناسب اندام عمومی باشد، مدت تمرین متوسط خواهد بود.

اگر وزن بیشتر از حد نرمال باشد و هدف افزایش عضله باشد، مدت تمرین طولانی خواهد بود.

اگر سن میانسال باشد و هدف کاهش وزن باشد، مدت تمرین متوسط خواهد بود.

```
# --- DURATION RULES ---
rule25 = ctrl.Rule(goal['weight_loss'] & weight['over'], duration['long'])
rule26 = ctrl.Rule(goal['weight_loss'] & energy['low'], duration['medium'])
rule27 = ctrl.Rule(goal['muscle_gain'], duration['medium'])
rule28 = ctrl.Rule(goal['general_fitness'] & energy['low'], duration['short'])
rule29 = ctrl.Rule(goal['general_fitness'] & energy['high'], duration['medium'])
rule30 = ctrl.Rule(fitness['advanced'] & energy['high'], duration['long'])
rule31 = ctrl.Rule(fitness['beginner'] & energy['low'], duration['short'])
rule32 = ctrl.Rule(age['elderly'], duration['short'])
rule33 = ctrl.Rule(age['middle'] & energy['medium'], duration['medium'])
rule34 = ctrl.Rule(age['young'] & energy['high'], duration['long'])
rule35 = ctrl.Rule(weight['under'] & goal['muscle_gain'], duration['medium'])
rule36 = ctrl.Rule(weight['normal'] & goal['general_fitness'], duration['medium'])
rule37 = ctrl.Rule(weight['over'] & energy['low'], duration['short'])
rule38 = ctrl.Rule(fitness['intermediate'] & goal['muscle_gain'], duration['medium'])
rule39 = ctrl.Rule(goal['weight_loss'] & fitness['advanced'], duration['long'])
rule40 = ctrl.Rule(goal['general_fitness'] & energy['medium'], duration['medium'])
rule41 = ctrl.Rule(age['elderly'] & energy['high'], duration['medium'])
rule42 = ctrl.Rule(fitness['advanced'] & goal['general_fitness'], duration['medium'])
rule43 = ctrl.Rule(weight['under'] & goal['weight_loss'], duration['short'])
rule44 = ctrl.Rule(age['young'] & energy['medium'], duration['medium'])
rule45 = ctrl.Rule(fitness['intermediate'] & goal['general_fitness'], duration['medium'])
rule46 = ctrl.Rule(weight['over'] & goal['muscle_gain'], duration['long'])
rule47 = ctrl.Rule(age['middle'] & goal['weight_loss'], duration['medium'])
```

### ۱-۵-۳- بازتعریف برخی قوانین براساس ترجیحات کاربر

برخی قوانین مربوط به شدت تمرین بر اساس خواسته و ترجیح کاربر دوباره تعریف می‌شوند. مثلاً اگر کاربری شدت تمرین را روی حالت راحت‌تر تنظیم کرد، برخی قوانین که در حالت عادی خروجی high یا medium داشتند، بازنویسی می‌شوند.

```
# Change some rules based on user preferences
def adjust_rules_based_on_preference(preference):
    global rule3, rule5, rule8, rule12, rule18, rule19, rule22

    if preference == "lighter":
        # Change rules with high or medium intensity to be lighter
        rule3 = ctrl.Rule(fitness['beginner'] & energy['high'], intensity['low'])
        rule5 = ctrl.Rule(fitness['intermediate'] & energy['medium'], intensity['low'])
        rule8 = ctrl.Rule(fitness['advanced'] & energy['medium'], intensity['medium'])
        rule12 = ctrl.Rule(goal['muscle_gain'] & energy['high'], intensity['medium'])
        rule18 = ctrl.Rule(goal['general_fitness'] & fitness['advanced'], intensity['medium'])
        rule19 = ctrl.Rule(goal['muscle_gain'] & fitness['intermediate'] & energy['medium'], intensity['medium'])
        rule22 = ctrl.Rule(weight['over'] & energy['high'] & goal['muscle_gain'], intensity['medium'])

    elif preference == "harder":
        # Change rules with low or medium intensity to be harder
        rule3 = ctrl.Rule(fitness['beginner'] & energy['high'], intensity['high'])
        rule5 = ctrl.Rule(fitness['intermediate'] & energy['medium'], intensity['high'])
        rule8 = ctrl.Rule(fitness['advanced'] & energy['medium'], intensity['high'])
        rule12 = ctrl.Rule(goal['muscle_gain'] & energy['high'], intensity['high'])
        rule18 = ctrl.Rule(goal['general_fitness'] & fitness['advanced'], intensity['high'])
        rule19 = ctrl.Rule(goal['muscle_gain'] & fitness['intermediate'] & energy['medium'], intensity['high'])
        rule22 = ctrl.Rule(weight['over'] & energy['high'] & goal['muscle_gain'], intensity['high'])

# User preferences
user_pref = input("Do you prefer 'lighter' or 'harder' workouts? (Type 'lighter', 'harder' or press Enter to skip): ").strip().lower()
if user_pref in ["lighter", "harder"]:
    adjust_rules_based_on_preference(user_pref)
```

### ۱-۶- استفاده از استنتاج فازی

در بخش control system، دو سیستم کنترل فازی (Fuzzy Control System) برای تعیین شدت تمرین (intensity) و مدت زمان تمرین (duration) تعریف می‌شوند. هر کدام از این سیستم‌ها مجموعه‌ای از قواعد فازی را به عنوان ورودی می‌گیرند.

- intensity\_ctrl: شامل قواعدی است که مشخص می‌کند شدت تمرین بر اساس ورودی‌هایی مانند سطح آمادگی جسمانی، انرژی، هدف تمرین، سن و وزن چگونه باشد.
- duration\_ctrl: شامل قواعدی است که مدت زمان تمرین را بر اساس همان ورودی‌ها تعیین می‌کند. هر ControlSystem مانند یک تصمیم‌گیرنده عمل می‌کند و با استفاده از قوانین فازی تعریف‌شده، خروجی مناسب را برحسب شرایط ارائه می‌دهد.

در بخش simulation، از هر سیستم کنترل تعریف‌شده، یک شیء شبیه‌سازی (Simulation Object) ساخته می‌شود. این اشیاء امکان دریافت ورودی‌های عددی (crisp) را فراهم کرده و پس از پردازش، خروجی‌های عددی نهایی را برای شدت و مدت تمرین ارائه می‌دهند.

```
# Control system
intensity_ctrl = ctrl.ControlSystem([rule1, rule2, rule3, rule4, rule5, rule6,
                                     rule7, rule8, rule9, rule10, rule11, rule12,
                                     rule13, rule14, rule15, rule16, rule17, rule18,
                                     rule19, rule20, rule21, rule22, rule23, rule24])

duration_ctrl = ctrl.ControlSystem([rule25, rule26, rule27, rule28, rule29,
                                    rule30, rule31, rule32, rule33, rule34,
                                    rule35, rule36, rule37, rule38, rule39,
                                    rule40, rule41, rule42, rule43, rule44,
                                    rule45, rule46, rule47])

# Simulators
intensity_sim = ctrl.ControlSystemSimulation(intensity_ctrl)
duration_sim = ctrl.ControlSystemSimulation(duration_ctrl)
```

## ۷-۱- تست و اعتبارسنجی سیستم

### ۷-۱-۱- تابع test\_system()

این تابع برای آزمایش و ارزیابی سیستم فازی طراحی شده است.

- fitness\_val: مقدار عددی برای سطح آمادگی جسمانی (از 0 تا 10)
- energy\_val: مقدار عددی سطح انرژی کاربر (از 0 تا 10)
- goal\_val: کد عددی هدف تمرین (0=کاهش وزن، 1=افزایش عضله، 2=تناسب عمومی)
- age\_val: سن فرد به صورت عددی
- weight\_val: وزن بدن فرد (برای تعیین کمبود وزن، نرمال یا اضافه وزن)

در ابتدا مقادری به ورودی‌های intensity\_sim و duration\_sim بر اساس پارامترهای ورودی انجام می‌شود. سپس با استفاده از compute()، سیستم فازی بر اساس قوانین موجود، مقدار خروجی‌های مربوط به شدت و مدت تمرین را محاسبه می‌کند. با استفاده از تابع categorize\_output() مقدار عددی به یکی از دسته‌های فازی مانند "کم"، "متوسط"، یا "زیاد" تبدیل می‌شود. (بیشینه درجه عضویت)

```
def test_system(fitness_val, energy_val, goal_val, age_val, weight_val):
    # Set input values
    intensity_sim.input['fitness'] = fitness_val
    intensity_sim.input['energy'] = energy_val
    intensity_sim.input['goal'] = goal_val
    intensity_sim.input['age'] = age_val
    intensity_sim.input['weight'] = weight_val

    duration_sim.input['fitness'] = fitness_val
    duration_sim.input['energy'] = energy_val
    duration_sim.input['goal'] = goal_val
    duration_sim.input['age'] = age_val
    duration_sim.input['weight'] = weight_val

    # Compute results
    intensity_sim.compute()
    duration_sim.compute()

    # Get crisp outputs
    intensity_value = intensity_sim.output['intensity']
    duration_value = duration_sim.output['duration']

    # Get fuzzy categories using membership functions
    intensity_cat = categorize_output(intensity_value, 'intensity')
    duration_cat = categorize_output(duration_value, 'duration')
```

#### ۱-۷-۲- تابع کمکی categorize\_output

تابع categorize\_output یک تابع کمکی در سیستم فازی است که مقدار عددی (crisp) خروجی تولیدشده توسط سیستم را به نزدیک‌ترین برچسب فازی ممکن (مانند «کم»، «متوسط» یا «زیاد» برای شدت، و «کوتاه»، «متوسط» یا «طولانی» برای مدت تمرین) تبدیل می‌کند. این تابع با استفاده از تابع interp\_membership میزان تعلق مقدار ورودی به هر تابع عضویت تعریف‌شده را محاسبه کرده و سپس برچسبی را انتخاب می‌کند که بیشترین درجه عضویت را دارد.

```
def categorize_output(value, variable):
    if variable == 'intensity':
        memberships = {
            'Low': fuzz.interp_membership(intensity.universe, intensity['low'].mf, value),
            'Medium': fuzz.interp_membership(intensity.universe, intensity['medium'].mf, value),
            'High': fuzz.interp_membership(intensity.universe, intensity['high'].mf, value)
        }
    elif variable == 'duration':
        memberships = {
            'Short': fuzz.interp_membership(duration.universe, duration['short'].mf, value),
            'Medium': fuzz.interp_membership(duration.universe, duration['medium'].mf, value),
            'Long': fuzz.interp_membership(duration.universe, duration['long'].mf, value)
        }
    else:
        raise ValueError("Unknown variable for categorization.")

    # Return the label with the highest degree of membership
    return max(memberships, key=memberships.get)
```

### ۱-۷-۳- تست کردن تابع test\_system()

```
test_system(2, 5, 1, 22, 68) # 1. Beginner, medium energy, muscle gain, young, normal weight
test_system(5, 2, 0, 45, 90) # 2. Intermediate, low energy, weight loss, middle-aged, overweight
test_system(9, 5, 2, 70, 45) # 3. Advanced, medium energy, general fitness, elderly, underweight
test_system(1, 9, 0, 50, 95) # 4. Beginner, high energy, weight loss, middle-aged, overweight
test_system(9, 1, 1, 28, 72) # 5. Advanced, low energy, muscle gain, young, normal weight
test_system(6, 9, 2, 30, 85) # 6. Intermediate, high energy, general fitness, young, overweight
test_system(1, 1, 2, 67, 48) # 7. Beginner, low energy, general fitness, elderly, underweight
test_system(5, 5, 1, 66, 70) # 8. Intermediate, medium energy, muscle gain, elderly, normal weight
test_system(8, 9, 0, 55, 100) # 9. Advanced, high energy, weight loss, middle-aged, overweight
test_system(3, 5, 2, 23, 49) # 10. Beginner, medium energy, general fitness, young, underweight
test_system(5, 2, 2, 68, 65) # 11. Intermediate, low energy, general fitness, elderly, normal weight
test_system(9, 6, 1, 29, 88) # 12. Advanced, medium energy, muscle gain, young, overweight
```

### خروجی های تست ها:

Test Case: Fitness: 9, Energy: 1, Goal: 1, Age: 28, Weight: 72  
Recommended Intensity: Medium (5.00)  
Recommended Duration: Medium (45.00)

Test Case: Fitness: 6, Energy: 9, Goal: 2, Age: 30, Weight: 85  
Recommended Intensity: High (6.74)  
Recommended Duration: Medium (54.72)

Test Case: Fitness: 1, Energy: 1, Goal: 2, Age: 67, Weight: 48  
Recommended Intensity: Low (1.56)  
Recommended Duration: Short (11.67)

Test Case: Fitness: 5, Energy: 5, Goal: 1, Age: 66, Weight: 70  
Recommended Intensity: Medium (5.32)  
Recommended Duration: Medium (29.62)

Test Case: Fitness: 8, Energy: 9, Goal: 0, Age: 55, Weight: 100  
Recommended Intensity: High (7.54)  
Recommended Duration: Long (67.37)

Test Case: Fitness: 3, Energy: 5, Goal: 2, Age: 23, Weight: 49  
Recommended Intensity: Low (1.76)  
Recommended Duration: Medium (45.00)

Test Case: Fitness: 5, Energy: 2, Goal: 2, Age: 68, Weight: 65  
Recommended Intensity: Low (2.93)  
Recommended Duration: Short (27.35)

Test Case: Fitness: 9, Energy: 6, Goal: 1, Age: 29, Weight: 88  
Recommended Intensity: High (8.24)  
Recommended Duration: Long (62.65)

Test Case: Fitness: 2, Energy: 5, Goal: 1, Age: 22, Weight: 68  
Recommended Intensity: Low (1.56)  
Recommended Duration: Medium (45.00)

Test Case: Fitness: 5, Energy: 2, Goal: 0, Age: 45, Weight: 90  
Recommended Intensity: Low (1.56)  
Recommended Duration: Medium (45.00)

Test Case: Fitness: 9, Energy: 5, Goal: 2, Age: 70, Weight: 45  
Recommended Intensity: Medium (5.00)  
Recommended Duration: Short (27.35)

Test Case: Fitness: 1, Energy: 9, Goal: 0, Age: 50, Weight: 95  
Recommended Intensity: Medium (5.00)  
Recommended Duration: Long (63.63)

Test Case: Fitness: 9, Energy: 1, Goal: 1, Age: 28, Weight: 72  
Recommended Intensity: Medium (5.00)  
Recommended Duration: Medium (45.00)

Test Case: Fitness: 6, Energy: 9, Goal: 2, Age: 30, Weight: 85  
Recommended Intensity: High (6.74)  
Recommended Duration: Medium (54.72)

Test Case: Fitness: 1, Energy: 1, Goal: 2, Age: 67, Weight: 48  
Recommended Intensity: Low (1.56)  
Recommended Duration: Short (11.67)

Test Case: Fitness: 5, Energy: 5, Goal: 1, Age: 66, Weight: 70  
Recommended Intensity: Medium (5.32)  
Recommended Duration: Medium (29.62)

## منابع

- [1] "chatgpt," [Online]. Available: <https://chatgpt.com/>.
- [2] "youtube," [Online]. Available:  
<https://www.youtube.com/watch?v=XuWPdFteIkQ>.
- [3] "youtube," [Online]. Available:  
<https://www.youtube.com/watch?v=XACvF3TtywM>.
- [4] Kruse, Mostaghim, Borgelt, Braune and Steinbrecher, Computational intelligence: A methodological introduction, 2022.