In [1]:

```
# Auto Reload Modules
%load_ext autoreload
%autoreload 2

import glob
import rosbag
import numpy as np
import os
import matplotlib.pyplot as plt
from tabulate import tabulate
```

Load rosbag

In [2]:

```
bag_files = glob.glob('*.bag')
print("Bag files:", bag_files)

b = rosbag.Bag(bag_files[0])
```

```
Bag files: ['2021-07-14-12-19-46.bag']
```

Display ROS topics and types

In [3]:

```
topics = b.get_type_and_topic_info().topics

topic_info = []

for key, topic_tuple in topics.items():
    topic_info.append([key, topic_tuple.frequency, topic_tuple.msg_type, topic_tuple.

print(tabulate(topic_info, headers=["Name", "Frequency", "Type", "Count"]))
```

```
Name                                       Frequency  Type
Count
------------------------------------  --------------  -------------------------------
-------
/diagnostics                              0.997254    diagnostic_msgs/DiagnosticArray
280
/game_control/joy                        35.968       sensor_msgs/Joy
3034
/pacmod/as_rx/accel_cmd                  36.0277      pacmod_msgs/SystemCmdFloat
3034
/pacmod/as_rx/brake_cmd                  36.0614      pacmod_msgs/SystemCmdFloat
3034
/pacmod/as_rx/shift_cmd                   0.00398089  pacmod_msgs/SystemCmdInt
2
/pacmod/as_rx/steer_cmd                  36.0416      pacmod_msgs/SteerSystemCmd
3034
/pacmod/as_tx/all_system_statuses        30.002       pacmod_msgs/AllSystemStatuses
9111
/pacmod/as_tx/enabled                    29.4215      std_msgs/Bool
8931
/pacmod/as_tx/vehicle_speed              29.4172      std_msgs/Float64
8928
/pacmod/can_rx                          880.786       can_msgs/Frame
55021
/pacmod/can_tx                          771.154       can_msgs/Frame
175043
```

```
/pacmod/parsed_tx/accel_aux_rpt           29.4207      pacmod_msgs/AccelAuxRpt
8929
/pacmod/parsed_tx/accel_rpt               29.4182      pacmod_msgs/SystemRptFloat
8929
/pacmod/parsed_tx/brake_aux_rpt           29.4221      pacmod_msgs/BrakeAuxRpt
8930
/pacmod/parsed_tx/brake_rpt               29.416       pacmod_msgs/SystemRptFloat
8930
/pacmod/parsed_tx/brake_rpt_detail_1      29.4215      pacmod_msgs/MotorRpt1
8928
/pacmod/parsed_tx/brake_rpt_detail_2      29.4188      pacmod_msgs/MotorRpt2
8929
/pacmod/parsed_tx/brake_rpt_detail_3      29.4186      pacmod_msgs/MotorRpt3
8928
/pacmod/parsed_tx/component_rpt           29.4193      pacmod_msgs/ComponentRpt
8929
/pacmod/parsed_tx/global_rpt              29.4203      pacmod_msgs/GlobalRpt
8930
/pacmod/parsed_tx/shift_aux_rpt           29.4488      pacmod_msgs/ShiftAuxRpt
8932
/pacmod/parsed_tx/shift_rpt               29.4468      pacmod_msgs/SystemRptInt
8933
/pacmod/parsed_tx/steer_rpt               29.4182      pacmod_msgs/SystemRptFloat
8929
/pacmod/parsed_tx/steer_rpt_detail_1      29.4153      pacmod_msgs/MotorRpt1
8926
/pacmod/parsed_tx/steer_rpt_detail_2      29.4149      pacmod_msgs/MotorRpt2
8926
/pacmod/parsed_tx/steer_rpt_detail_3      29.4168      pacmod_msgs/MotorRpt3
8926
/pacmod/parsed_tx/vehicle_speed_rpt       29.4172      pacmod_msgs/VehicleSpeedRpt
8928
/rosout                                47662.5         rosgraph_msgs/Log
20
```

Topic Analysis

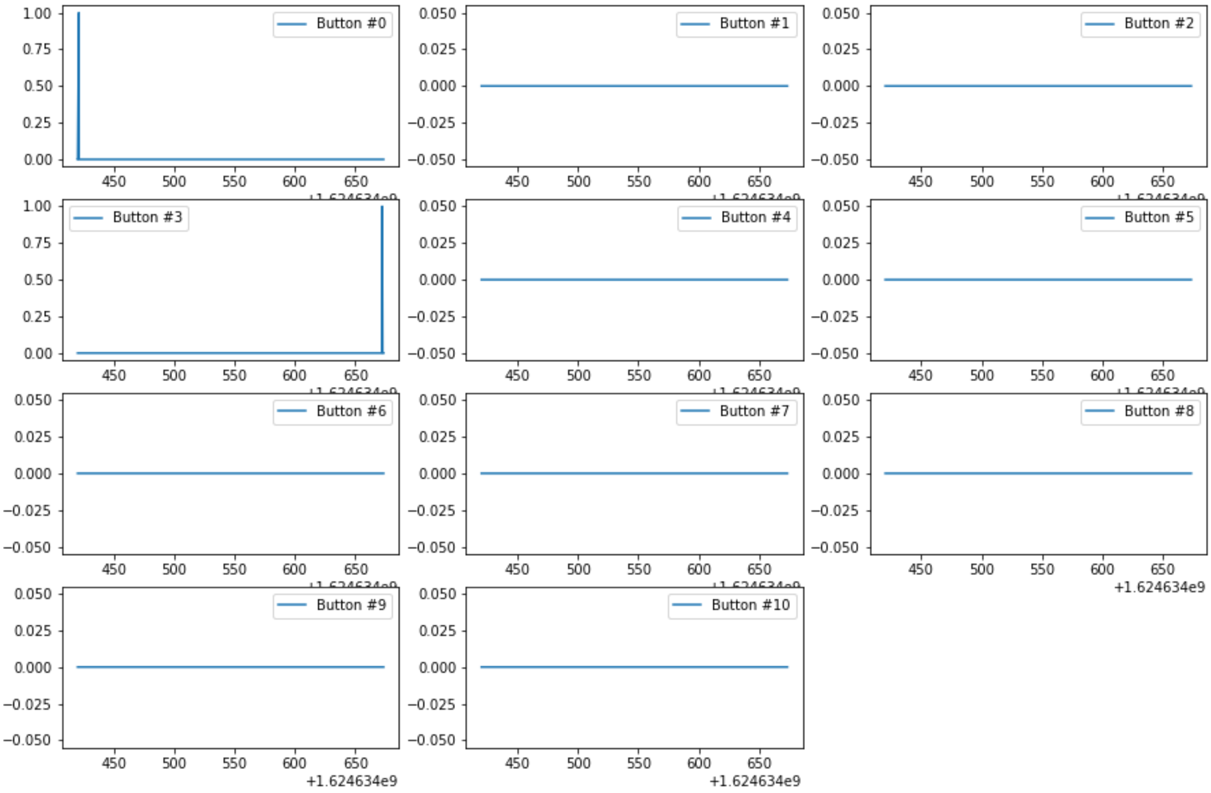
/game_control/joy

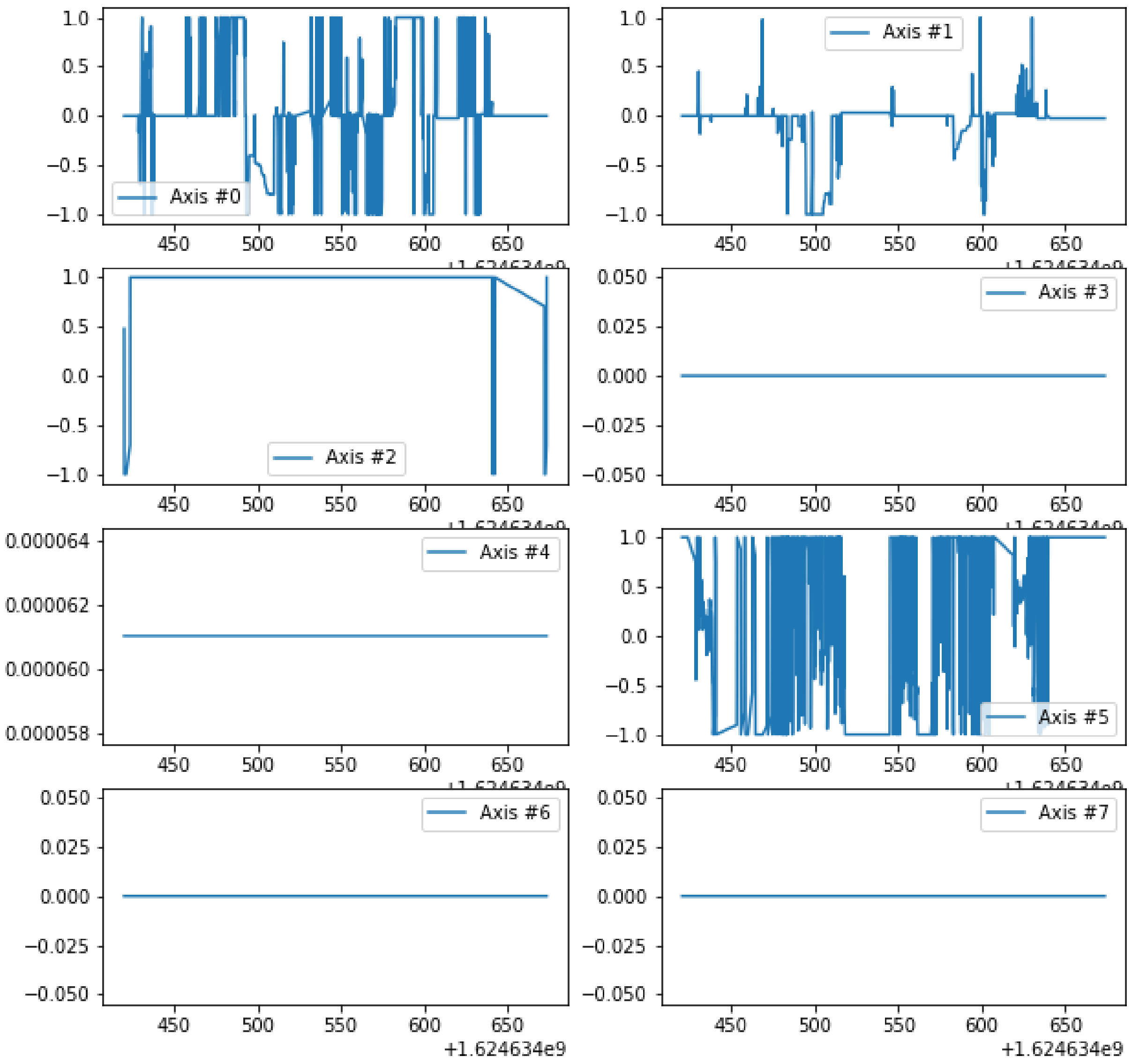
In [4]:

```python
time = []
buttons = []
axes = []
for _, msg, t in b.read_messages('/game_control/joy'):
    time.append(t.to_sec())
    axes.append(msg.axes)
    buttons.append(msg.buttons)

print("====== Message example ======")
print(msg)

buttons = np.array(buttons)
axes = np.array(axes)

f1 = plt.figure(figsize=(15,10))

for i in range(len(buttons[0])):
    ax = plt.subplot(4,3,i+1)
    ax.plot(time, buttons[:, i], label="Button #%d" % i)
    ax.legend()

f2 = plt.figure(figsize=(10,10))
```

```
for i in range(len(axes[0])):
    ax = plt.subplot(4,2,i+1)
    ax.plot(time, axes[:, i], label=″Axis #%d″ % i)
    ax.legend()
```

```
====== Message example ======
header:
  seq: 19696
  stamp:
    secs: 1624634673
    nsecs: 559517337
  frame_id: ″/dev/input/js0″
axes: [-0.0, -0.023712882772088805, 1.0, -0.0, 6.103701889514923e-05, 1.0, -0.0, -0.0]
buttons: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```





/pacmod/as_rx/accel_cmd

In [5]:

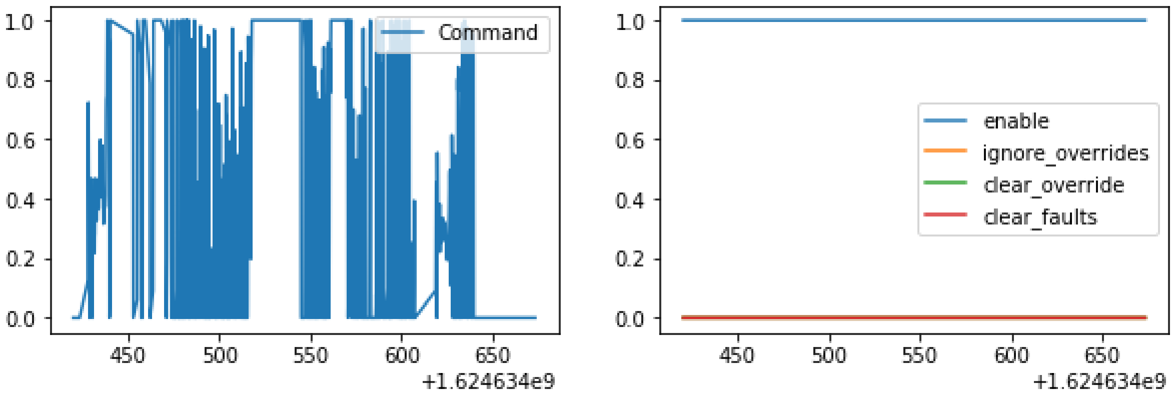
```
time = []
command = []
flags = {
    ″enable″: [],
    ″ignore_overrides″: [],
    ″clear_override″: [],
    ″clear_faults″: []
}
for _, msg, t in b. read_messages(’/pacmod/as_rx/accel_cmd’):
    time. append(t. to_sec())
    command. append(msg. command)
    flags[″enable″]. append(msg. enable)
    flags[″ignore_overrides″]. append(msg. ignore_overrides)
    flags[″clear_override″]. append(msg. clear_override)
    flags[″clear_faults″]. append(msg. clear_faults)

print(″====== Message example ======″)
print(msg)

command = np. array(command)

f = plt. figure(figsize=(10, 3))

ax1 = plt. subplot(1,2,1)
```

```
ax1.plot(time, command, label="Command")
ax1.legend()

ax2 = plt.subplot(1,2,2)
for name, value in flags.items():
    ax2.plot(time, value, label=name)

ax2.legend()
```

```
====== Message example ======
header:
  seq: 19694
  stamp:
    secs: 0
    nsecs:         0
  frame_id: ''
enable: True
ignore_overrides: False
clear_override: False
clear_faults: False
command: -0.0
```

Out[5]:

```
<matplotlib.legend.Legend at 0x7f6d337d4d00>
```



/pacmod/as_rx/brake_cmd

In [6]:

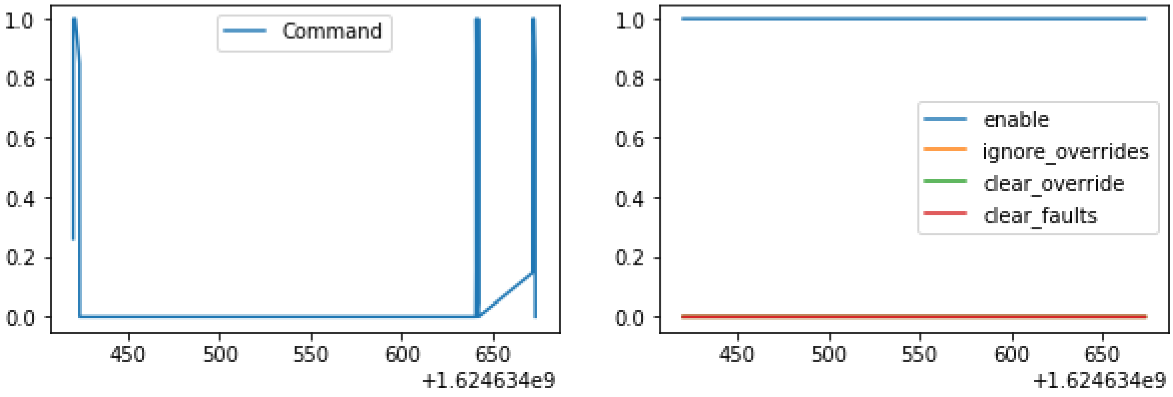
```
time = []
command = []
flags = {
    "enable": [],
    "ignore_overrides": [],
    "clear_override": [],
    "clear_faults": []
}
for _, msg, t in b.read_messages('/pacmod/as_rx/brake_cmd'):
    time.append(t.to_sec())
    command.append(msg.command)
    flags["enable"].append(msg.enable)
    flags["ignore_overrides"].append(msg.ignore_overrides)
    flags["clear_override"].append(msg.clear_override)
    flags["clear_faults"].append(msg.clear_faults)

print("====== Message example ======")
print(msg)

command = np.array(command)

f = plt.figure(figsize=(10, 3))
```

```
ax1 = plt.subplot(1,2,1)
ax1.plot(time, command, label="Command")
ax1.legend()

ax2 = plt.subplot(1,2,2)
for name, value in flags.items():
    ax2.plot(time, value, label=name)

ax2.legend()
```

```
====== Message example ======
header:
  seq: 19694
  stamp:
    secs: 0
    nsecs:          0
  frame_id: ''
enable: True
ignore_overrides: False
clear_override: False
clear_faults: False
command: -0.0
```

Out[6]:

```
<matplotlib.legend.Legend at 0x7f6d2b1a8f10>
```



/pacmod/as_rx/shift_cmd

In [7]:

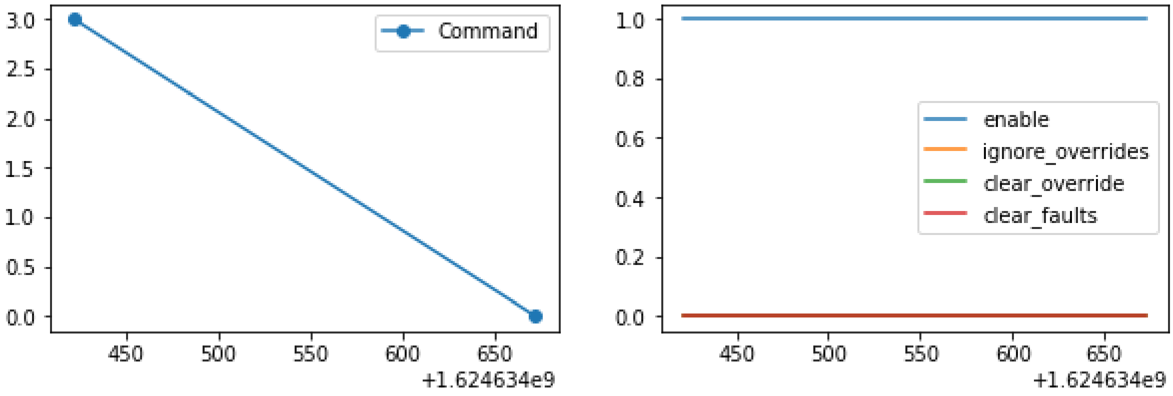
```
time = []
command = []
flags = {
    "enable": [],
    "ignore_overrides": [],
    "clear_override": [],
    "clear_faults": []
}

for _, msg, t in b.read_messages('/pacmod/as_rx/shift_cmd'):
    time.append(t.to_sec())
    command.append(msg.command)
    flags["enable"].append(msg.enable)
    flags["ignore_overrides"].append(msg.ignore_overrides)
    flags["clear_override"].append(msg.clear_override)
    flags["clear_faults"].append(msg.clear_faults)

print("====== Message example ======")
print(msg)

command = np.array(command)
```

```
f = plt. figure(figsize=(10,  3))

ax1 = plt. subplot(1, 2, 1)
ax1. plot(time,  command,  marker=’o’,  label=”Command”)
ax1. legend()

ax2 = plt. subplot(1, 2, 2)
for name,  value in flags. items():
    ax2. plot(time,  value,  label=name)

ax2. legend()
```

```
====== Message example ======
header:
  seq: 42
  stamp:
    secs: 0
    nsecs:         0
  frame_id: ''
enable: True
ignore_overrides: False
clear_override: False
clear_faults: False
command: 0
```

Out[7]:

```
<matplotlib. legend. Legend at 0x7f6d2b149a30>
```



/pacmod/as_rx/steer_cmd

In [8]:

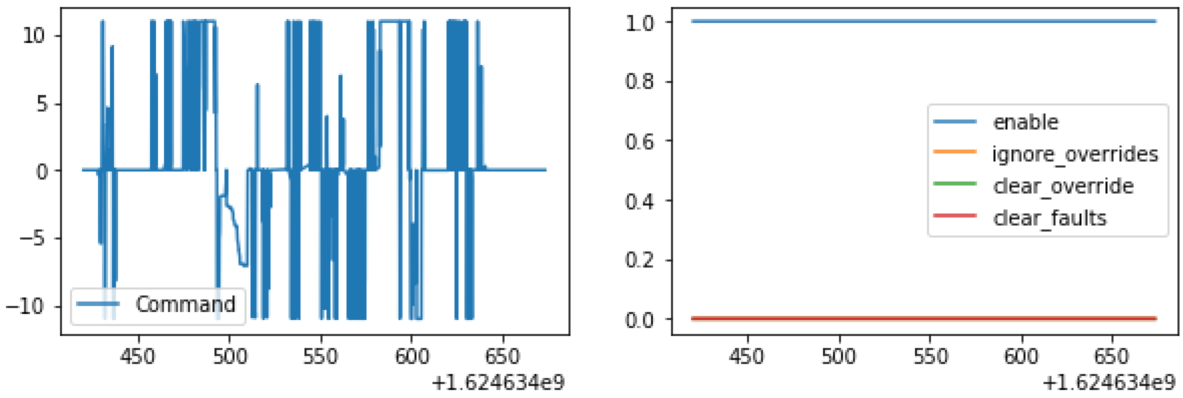
```
time = []
command = []
flags = {
    ”enable”: [],
    ”ignore_overrides”: [],
    ”clear_override”: [],
    ”clear_faults”: []
}
for _,  msg,  t in b. read_messages(’/pacmod/as_rx/steer_cmd’):
    time. append(t. to_sec())
    command. append(msg. command)
    flags[”enable”]. append(msg. enable)
    flags[”ignore_overrides”]. append(msg. ignore_overrides)
    flags[”clear_override”]. append(msg. clear_override)
    flags[”clear_faults”]. append(msg. clear_faults)

print(”====== Message example ======”)
print(msg)

command = np. array(command)
```

```
f = plt.figure(figsize=(10, 3))

ax1 = plt.subplot(1,2,1)
ax1.plot(time, command, label="Command")
ax1.legend()

ax2 = plt.subplot(1,2,2)
for name, value in flags.items():
    ax2.plot(time, value, label=name)

ax2.legend()
```

```
====== Message example ======
header:
  seq: 19694
  stamp:
    secs: 0
    nsecs:         0
  frame_id: ''
enable: True
ignore_overrides: False
clear_override: False
clear_faults: False
command: -0.0
rotation_rate: 3.3
```

Out[8]:

```
<matplotlib.legend.Legend at 0x7f6d2b0bb7f0>
```



/pacmod/as_tx/all_system_statuses

In [9]:

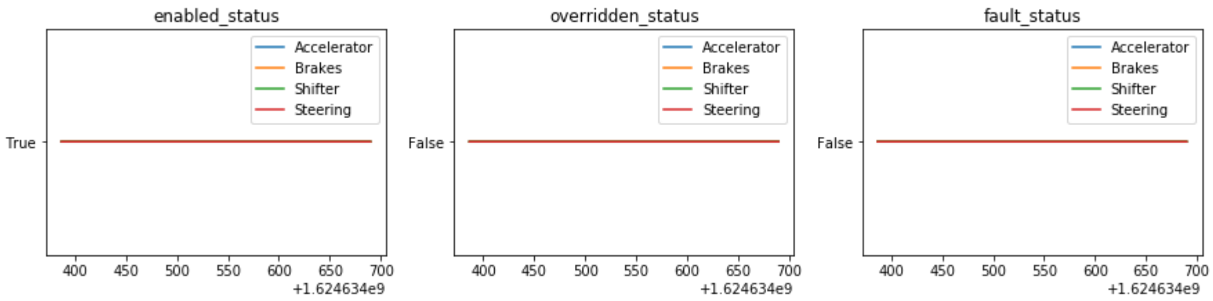
```
time = []
enabled_status = {
    "Accelerator": [],
    "Brakes": [],
    "Shifter": [],
    "Steering": []
}
overridden_status = {
    "Accelerator": [],
    "Brakes": [],
    "Shifter": [],
    "Steering": []
}
fault_status = {
    "Accelerator": [],
    "Brakes": [],
    "Shifter": [],
    "Steering": []
```

```python
}

for _, msg, t in b.read_messages('/pacmod/as_tx/all_system_statuses'):
    time.append(t.to_sec())

    enabled_status["Accelerator"].append(msg.enabled_status[0].value)
    enabled_status["Brakes"].append(msg.enabled_status[1].value)
    enabled_status["Shifter"].append(msg.enabled_status[2].value)
    enabled_status["Steering"].append(msg.enabled_status[3].value)

    overridden_status["Accelerator"].append(msg.overridden_status[0].value)
    overridden_status["Brakes"].append(msg.overridden_status[1].value)
    overridden_status["Shifter"].append(msg.overridden_status[2].value)
    overridden_status["Steering"].append(msg.overridden_status[3].value)

    fault_status["Accelerator"].append(msg.fault_status[0].value)
    fault_status["Brakes"].append(msg.fault_status[1].value)
    fault_status["Shifter"].append(msg.fault_status[2].value)
    fault_status["Steering"].append(msg.fault_status[3].value)

print("====== Message example ======")
print(msg)

# enabled_status = np.array(enabled_status)
# overridden_status = np.array(overridden_status)
# fault_status = np.array(fault_status)

f = plt.figure(figsize=(15, 3))

ax1 = plt.subplot(1,3,1)
ax1.set_title("enabled_status")
for name, value in enabled_status.items():
    ax1.plot(time, value, label=name)
ax1.legend()

ax2 = plt.subplot(1,3,2)
ax2.set_title("overridden_status")
for name, value in overridden_status.items():
    ax2.plot(time, value, label=name)
ax2.legend()

ax3 = plt.subplot(1,3,3)
ax3.set_title("fault_status")
for name, value in fault_status.items():
    ax3.plot(time, value, label=name)
ax3.legend()
```

```
====== Message example ======
header:
  seq: 79734
  stamp:
    secs: 0
    nsecs:         0
  frame_id: ''
enabled_status:
  -
    key: "Accelerator"
    value: "True"
  -
    key: "Brakes"
    value: "True"
  -
    key: "Shifter"
    value: "True"
  -
    key: ”Steering”
    value: ”True”
overridden_status:
  -
    key: ”Accelerator”
    value: ”False”
  -
    key: ”Brakes”
    value: ”False”
  -
    key: ”Shifter”
    value: ”False”
  -
    key: ”Steering”
    value: ”False”
fault_status:
  -
    key: ”Accelerator”
    value: ”False”
  -
    key: ”Brakes”
    value: ”False”
  -
    key: ”Shifter”
    value: ”False”
  -
    key: ”Steering”
    value: ”False”
```

Out[9]:

```
<matplotlib.legend.Legend at 0x7f6d2a614e80>
```



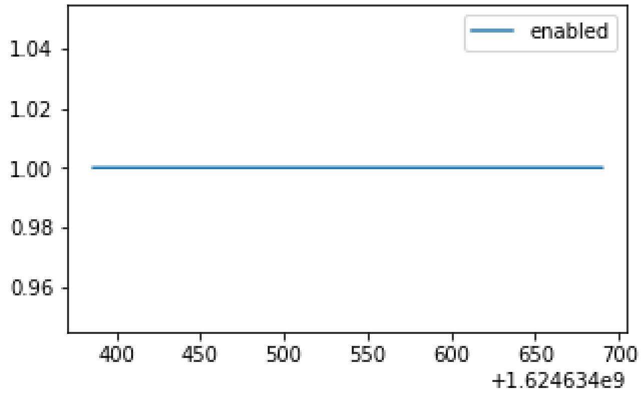
/pacmod/as_tx/enabled

In [10]:

```
time = []
flags = []
for _, msg, t in b.read_messages(’/pacmod/as_tx/enabled’):
    time.append(t.to_sec())
    flags.append(msg.data)

print(”====== Message example ======”)
print(msg)

flags = np.array(flags)

f = plt.figure(figsize=(5, 3))

plt.plot(time, flags, label=”enabled”)
plt.legend()
```

```
====== Message example ======
data: True
<matplotlib.legend.Legend at 0x7f6d2b0468e0>
```

Out[10]:



/pacmod/as_tx/vehicle_speed

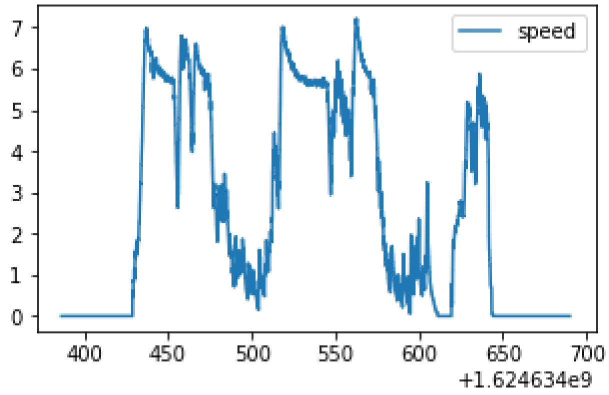
In [11]:

```
time = []
speed = []
for _, msg, t in b.read_messages('/pacmod/as_tx/vehicle_speed'):
    time.append(t.to_sec())
    speed.append(msg.data)

print("====== Message example ======")
print(msg)

speed = np.array(speed)

f = plt.figure(figsize=(5, 3))

plt.plot(time, speed, label="speed")
plt.legend()
```

```
====== Message example ======
data: 0.0
```

Out[11]:

```
<matplotlib.legend.Legend at 0x7f6d2a4daf10>
```



/pacmod/can_rx & /pacmod/can_tx

In [12]:

```
print("======== Message example in /pacmod/can_rx ========")
for _, msg, t in b.read_messages('/pacmod/can_rx'):
    print(msg)
    break

print("======== Message example in /pacmod/can_tx ========")
for _, msg, t in b.read_messages('/pacmod/can_tx'):
    print(msg)
    break
```

```
======== Message example in /pacmod/can_rx ========
header:
  seq: 426090
  stamp:
    secs: 1624634386
    nsecs: 337870468
  frame_id: ''
id: 320
is_rtr: False
is_extended: False
is_error: False
dlc: 2
data: [0, 0, 0, 0, 0, 0, 0, 0]
======== Message example in /pacmod/can_tx ========
header:
  seq: 1356853
  stamp:
    secs: 1624634386
    nsecs: 312822169
  frame_id: "0"
id: 34
is_rtr: False
is_extended: False
is_error: False
dlc: 6
data: [2, 0, 8, 0, 60, 0, 0, 0]
```

/pacmod/parsed_tx/accel_aux_rpt

In [13]:

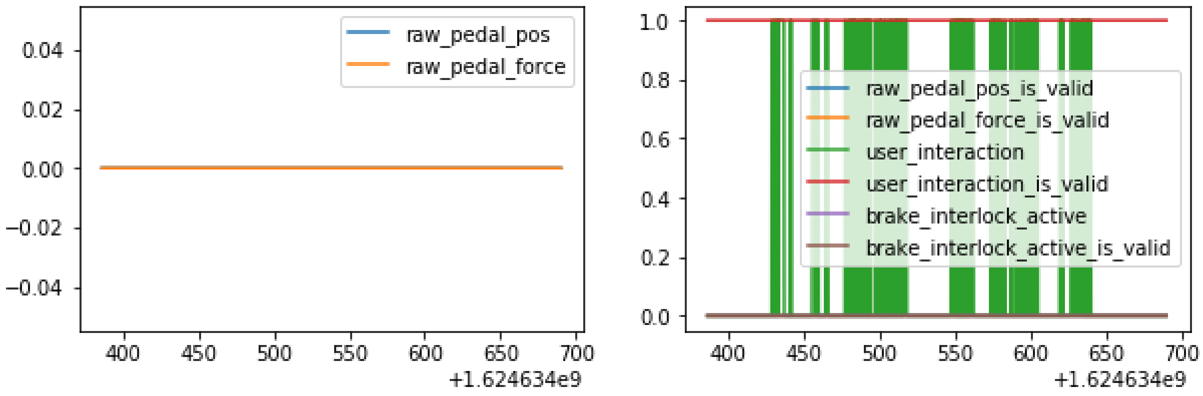
```
time = []
raw_pedal_pos = []
raw_pedal_force = []
flags = {"raw_pedal_pos_is_valid": [],
        "raw_pedal_force_is_valid": [],
        "user_interaction": [],
        "user_interaction_is_valid": [],
        "brake_interlock_active": [],
        "brake_interlock_active_is_valid": []}

for _, msg, t in b.read_messages('/pacmod/parsed_tx/accel_aux_rpt'):
    time.append(t.to_sec())
    raw_pedal_pos.append(msg.raw_pedal_pos)
    raw_pedal_force.append(msg.raw_pedal_force)
    flags["raw_pedal_pos_is_valid"].append(msg.raw_pedal_pos_is_valid)
    flags["raw_pedal_force_is_valid"].append(msg.raw_pedal_force_is_valid)
    flags["user_interaction"].append(msg.user_interaction)
    flags["user_interaction_is_valid"].append(msg.user_interaction_is_valid)
    flags["brake_interlock_active"].append(msg.brake_interlock_active)
    flags["brake_interlock_active_is_valid"].append(msg.brake_interlock_active_is_val

print("====== Message example ======")
print(msg)

f = plt.figure(figsize=(10, 3))

ax1 = plt.subplot(1,2,1)
ax1.plot(time, raw_pedal_pos, label="raw_pedal_pos")
ax1.plot(time, raw_pedal_force, label="raw_pedal_force")
ax1.legend()
```

```
ax2 = plt.subplot(1,2,2)
for key, value in flags.items():
    ax2.plot(time, value, label=key)
ax2.legend()
```

```
====== Message example ======
header:
  seq: 78138
  stamp:
    secs: 1624634689
    nsecs: 998117898
  frame_id: "pacmod"
raw_pedal_pos: 0.0
raw_pedal_pos_is_valid: False
raw_pedal_force: 0.0
raw_pedal_force_is_valid: False
user_interaction: False
user_interaction_is_valid: True
brake_interlock_active: False
brake_interlock_active_is_valid: False
```

Out[13]:

```
<matplotlib.legend.Legend at 0x7f6d2a3f62b0>
```



/pacmod/parsed_tx/accel_rpt

In [14]:

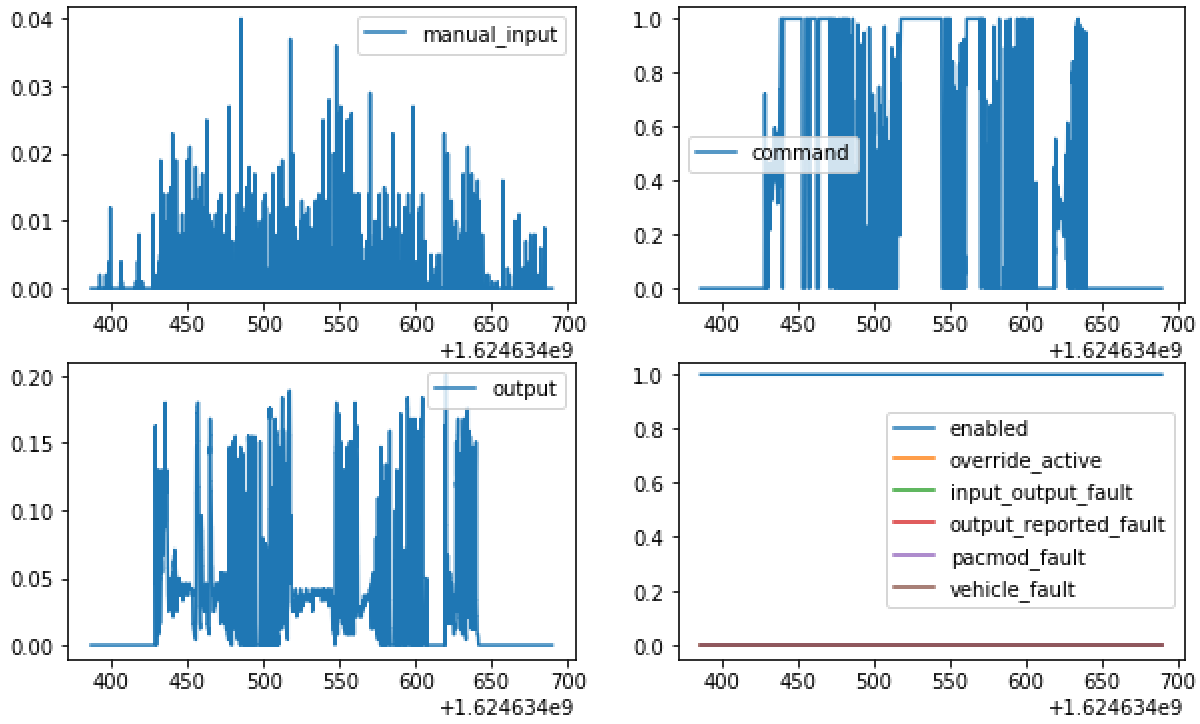
```
time = []
manual_input = []
command = []
output = []
flags = {"enabled": [],
        "override_active": [],
        "input_output_fault": [],
        "output_reported_fault": [],
        "pacmod_fault": [],
        "vehicle_fault": []}

for _, msg, t in b.read_messages('/pacmod/parsed_tx/accel_rpt'):
    time.append(t.to_sec())
    manual_input.append(msg.manual_input)
    command.append(msg.command)
    output.append(msg.output)

    flags["enabled"].append(msg.enabled)
    flags["override_active"].append(msg.override_active)
    flags["input_output_fault"].append(msg.input_output_fault)
    flags["output_reported_fault"].append(msg.output_reported_fault)
    flags["pacmod_fault"].append(msg.pacmod_fault)
    flags["vehicle_fault"].append(msg.vehicle_fault)
```

```
print("====== Message example ======")
print(msg)

f = plt.figure(figsize=(10, 6))

ax1 = plt.subplot(2,2,1)
ax1.plot(time, manual_input, label="manual_input")
ax1.legend()

ax2 = plt.subplot(2,2,2)
ax2.plot(time, command, label="command")
ax2.legend()

ax3 = plt.subplot(2,2,3)
ax3.plot(time, output, label="output")
ax3.legend()

ax4 = plt.subplot(2,2,4)
for key, value in flags.items():
    ax4.plot(time, value, label=key)
ax4.legend()
```

```
====== Message example ======
header:
  seq: 78139
  stamp:
    secs: 1624634689
    nsecs: 996658930
  frame_id: "pacmod"
enabled: True
override_active: False
command_output_fault: False
input_output_fault: False
output_reported_fault: False
pacmod_fault: False
vehicle_fault: False
manual_input: 0.0
command: 0.0
output: 0.0
```

Out[14]:

```
<matplotlib.legend.Legend at 0x7f6d2a162220>
```



/pacmod/parsed_tx/brake_aux_rpt

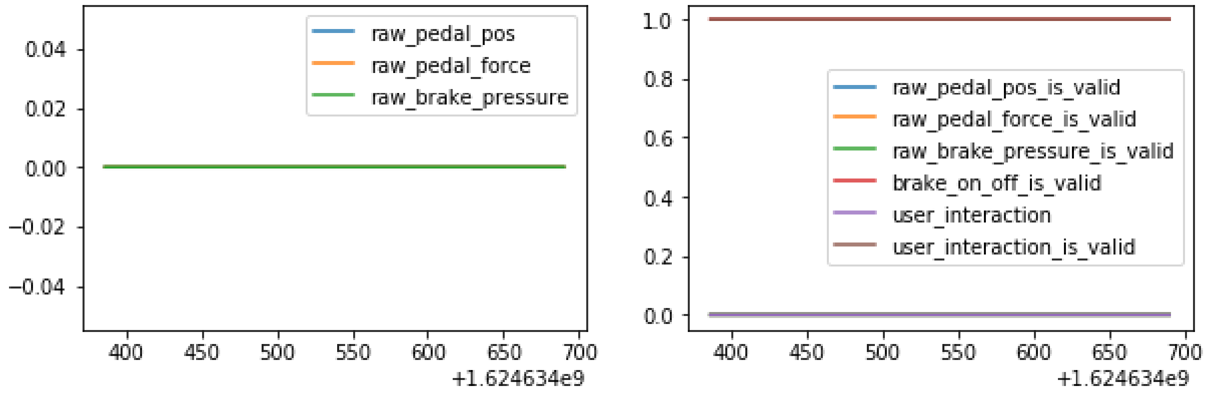
In [15]:

```
time = []
raw_pedal_pos = []
raw_pedal_force = []
raw_brake_pressure = []
flags = {"raw_pedal_pos_is_valid": [],
        "raw_pedal_force_is_valid": [],
        "raw_brake_pressure_is_valid": [],
        "brake_on_off_is_valid": [],
        "user_interaction": [],
        "user_interaction_is_valid": []}

for _, msg, t in b.read_messages('/pacmod/parsed_tx/brake_aux_rpt'):
    time.append(t.to_sec())

    raw_pedal_pos.append(msg.raw_pedal_pos)
    raw_pedal_force.append(msg.raw_pedal_force)
    raw_brake_pressure.append(msg.raw_brake_pressure)

    flags["raw_pedal_pos_is_valid"].append(msg.raw_pedal_pos_is_valid)
    flags["raw_pedal_force_is_valid"].append(msg.raw_pedal_force_is_valid)
    flags["raw_brake_pressure_is_valid"].append(msg.raw_brake_pressure_is_valid)
    flags["brake_on_off_is_valid"].append(msg.brake_on_off_is_valid)
    flags["user_interaction"].append(msg.user_interaction)
    flags["user_interaction_is_valid"].append(msg.user_interaction_is_valid)

print("====== Message example ======")
print(msg)

f = plt.figure(figsize=(10, 3))

ax1 = plt.subplot(1,2,1)
ax1.plot(time, raw_pedal_pos, label="raw_pedal_pos")
ax1.plot(time, raw_pedal_force, label="raw_pedal_force")
ax1.plot(time, raw_brake_pressure, label="raw_brake_pressure")
ax1.legend()

ax2 = plt.subplot(1,2,2)
for key, value in flags.items():
    ax2.plot(time, value, label=key)
ax2.legend()
```

```
====== Message example ======
header:
  seq: 78135
  stamp:
    secs: 1624634690
    nsecs:   4319420
  frame_id: "pacmod"
raw_pedal_pos: 0.0
raw_pedal_pos_is_valid: False
raw_pedal_force: 0.0
raw_pedal_force_is_valid: False
raw_brake_pressure: 0.0
raw_brake_pressure_is_valid: False
brake_on_off: False
brake_on_off_is_valid: True
user_interaction: False
user_interaction_is_valid: True
```

Out[15]:

```
<matplotlib.legend.Legend at 0x7f6d2a06ea60>
```



/pacmod/parsed_tx/brake_rpt

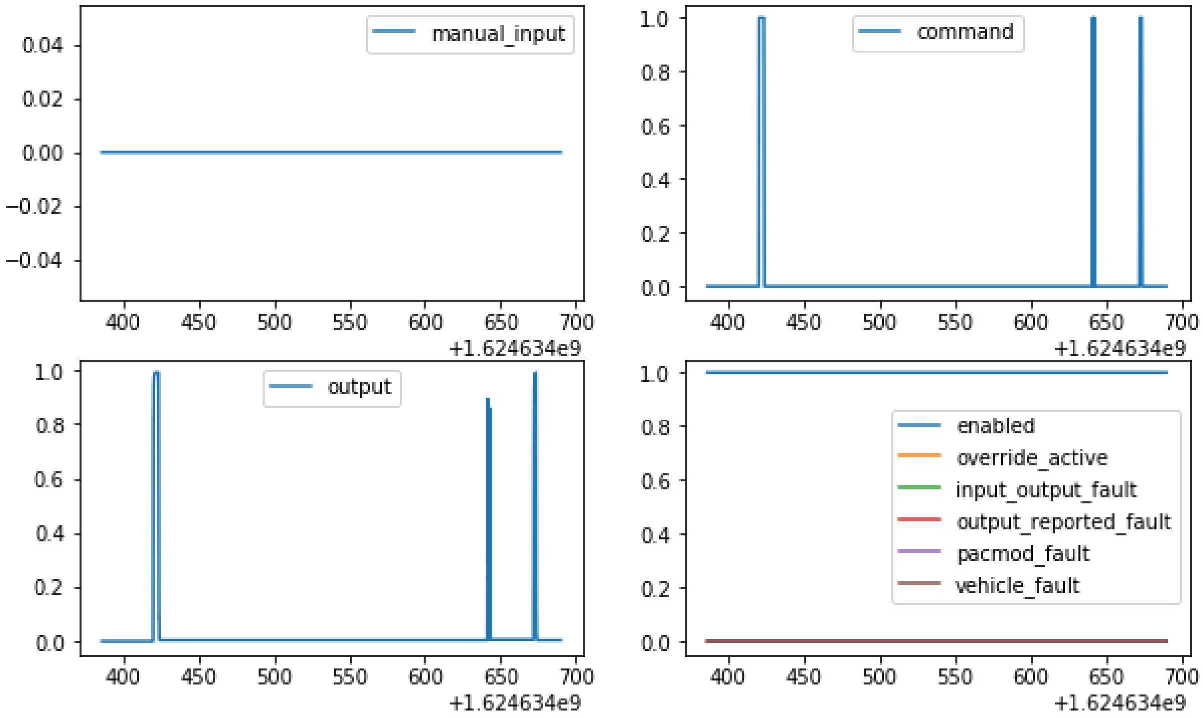
In [16]:

```
time = []
manual_input = []
command = []
output = []
flags = {"enabled": [],
        "override_active": [],
        "input_output_fault": [],
        "output_reported_fault": [],
        "pacmod_fault": [],
        "vehicle_fault": []}

for _, msg, t in b.read_messages('/pacmod/parsed_tx/brake_rpt'):
    time.append(t.to_sec())

    manual_input.append(msg.manual_input)
    command.append(msg.command)
    output.append(msg.output)

    flags["enabled"].append(msg.enabled)
    flags["override_active"].append(msg.override_active)
    flags["input_output_fault"].append(msg.input_output_fault)
    flags["output_reported_fault"].append(msg.output_reported_fault)
    flags["pacmod_fault"].append(msg.pacmod_fault)
    flags["vehicle_fault"].append(msg.vehicle_fault)

print("====== Message example ======")
print(msg)

f = plt.figure(figsize=(10, 6))

ax1 = plt.subplot(2,2,1)
ax1.plot(time, manual_input, label="manual_input")
ax1.legend()

ax2 = plt.subplot(2,2,2)
ax2.plot(time, command, label="command")
ax2.legend()

ax3 = plt.subplot(2,2,3)
ax3.plot(time, output, label="output")
ax3.legend()

ax4 = plt.subplot(2,2,4)
for key, value in flags.items():
    ax4.plot(time, value, label=key)
ax4.legend()
```

```
====== Message example ======
header:
  seq: 78136
  stamp:
    secs: 1624634689
    nsecs: 999491152
  frame_id: ″pacmod″
enabled: True
override_active: False
command_output_fault: False
input_output_fault: False
output_reported_fault: False
pacmod_fault: False
vehicle_fault: False
manual_input: 0.0
command: 0.0
output: 0.004
```

Out[16]:

```
<matplotlib.legend.Legend at 0x7f6d29eff100>
```



/pacmod/parsed_tx/brake_rpt_detail_1

In [17]:

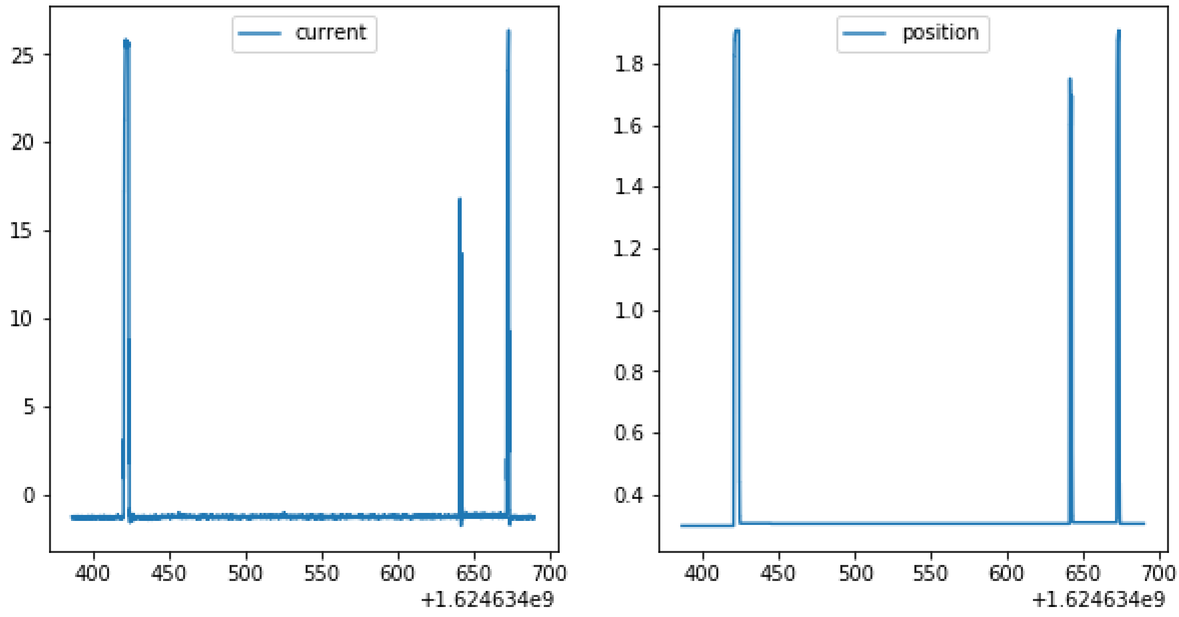
```
time = []
current = []
position = []

for _, msg, t in b.read_messages(’/pacmod/parsed_tx/brake_rpt_detail_1’):
    time.append(t.to_sec())
    current.append(msg.current)
    position.append(msg.position)

print(″====== Message example ======″)
print(msg)

f = plt.figure(figsize=(10, 5))

ax1 = plt.subplot(1,2,1)
ax1.plot(time, current, label=″current″)
ax1.legend()
```

```
ax2 = plt.subplot(1,2,2)
ax2.plot(time, position, label="position")
ax2.legend()
```

```
====== Message example ======
header:
  seq: 78126
  stamp:
    secs: 1624634689
    nsecs: 987192909
  frame_id: "pacmod"
current: -1.421
position: 0.306
```

Out[17]:

```
<matplotlib.legend.Legend at 0x7f6d29dc9f70>
```



/pacmod/parsed_tx/brake_rpt_detail_2

In [18]:

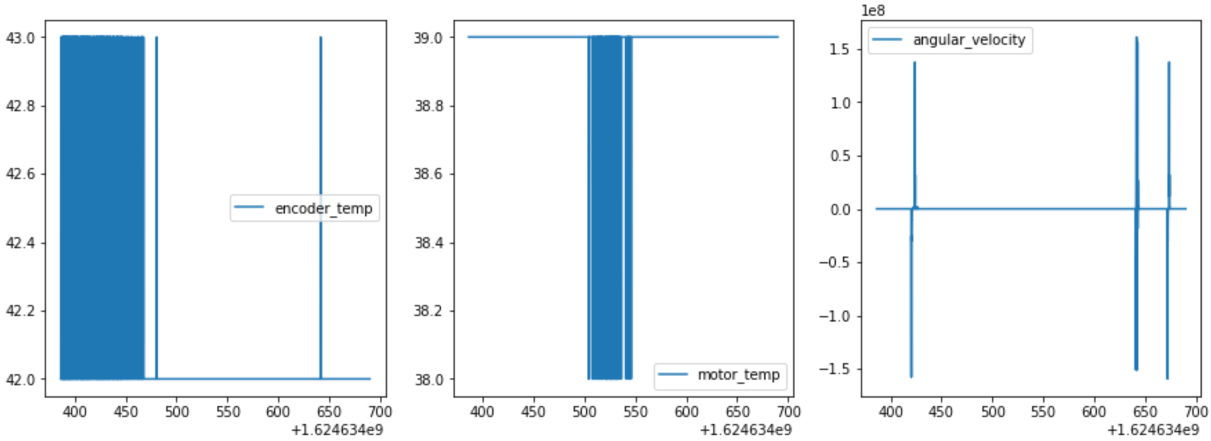
```
time = []
encoder_temp = []
motor_temp = []
angular_velocity = []

for _, msg, t in b.read_messages('/pacmod/parsed_tx/brake_rpt_detail_2'):
    time.append(t.to_sec())

    encoder_temp.append(msg.encoder_temp)
    motor_temp.append(msg.motor_temp)
    angular_velocity.append(msg.angular_velocity)

print("====== Message example ======")
print(msg)

f = plt.figure(figsize=(15, 5))

ax1 = plt.subplot(1,3,1)
ax1.plot(time, encoder_temp, label="encoder_temp")
ax1.legend()

ax2 = plt.subplot(1,3,2)
ax2.plot(time, motor_temp, label="motor_temp")
ax2.legend()
```

```
ax3 = plt.subplot(1,3,3)
ax3.plot(time, angular_velocity, label="angular_velocity")
ax3.legend()
```

```
====== Message example ======
header:
  seq: 78125
  stamp:
    secs: 1624634689
    nsecs: 988315981
  frame_id: "pacmod"
encoder_temp: 42.0
motor_temp: 39.0
angular_velocity: 0.0
```

Out[18]:

```
<matplotlib.legend.Legend at 0x7f6d29bcd280>
```



/pacmod/parsed_tx/brake_rpt_detail_3

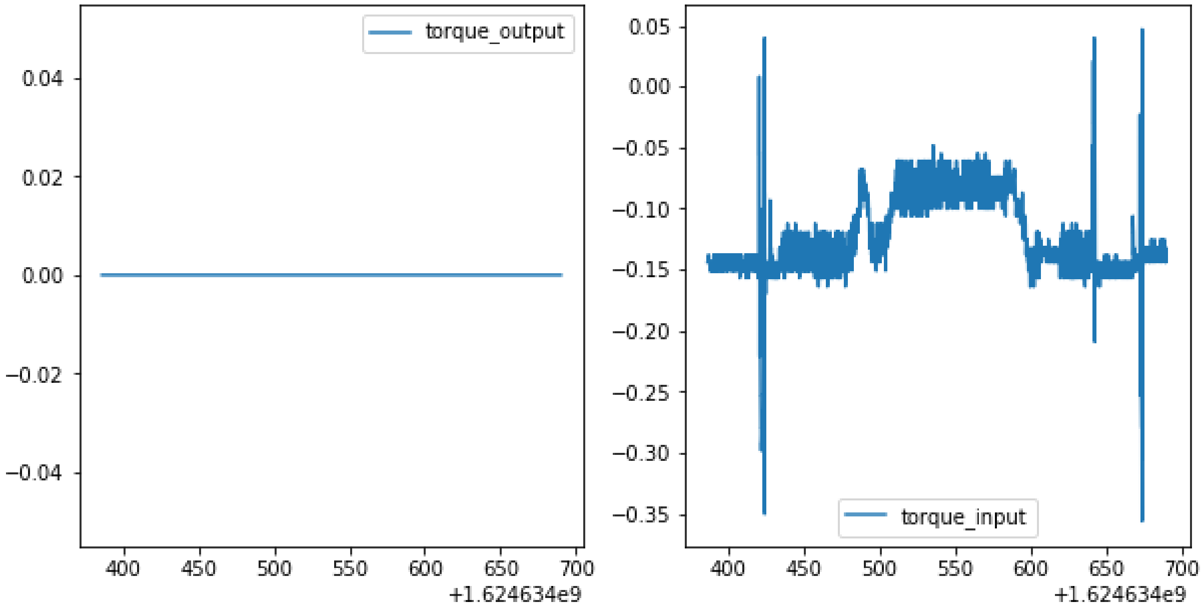
In [19]:

```
time = []
torque_output = []
torque_input = []

for _, msg, t in b.read_messages('/pacmod/parsed_tx/brake_rpt_detail_3'):
    time.append(t.to_sec())

    torque_output.append(msg.torque_output)
    torque_input.append(msg.torque_input)

print("====== Message example ======")
print(msg)

f = plt.figure(figsize=(10, 5))

ax1 = plt.subplot(1,2,1)
ax1.plot(time, torque_output, label="torque_output")
ax1.legend()

ax2 = plt.subplot(1,2,2)
ax2.plot(time, torque_input, label="torque_input")
ax2.legend()
```

```
====== Message example ======
header:
  seq: 78121
  stamp:
    secs: 1624634690
    nsecs:    416567
  frame_id: "pacmod"
torque_output: 0.0
torque_input: -0.138
```

Out[19]:

```
<matplotlib.legend.Legend at 0x7f6d29a24fa0>
```



/pacmod/parsed_tx/component_rpt

In [20]:

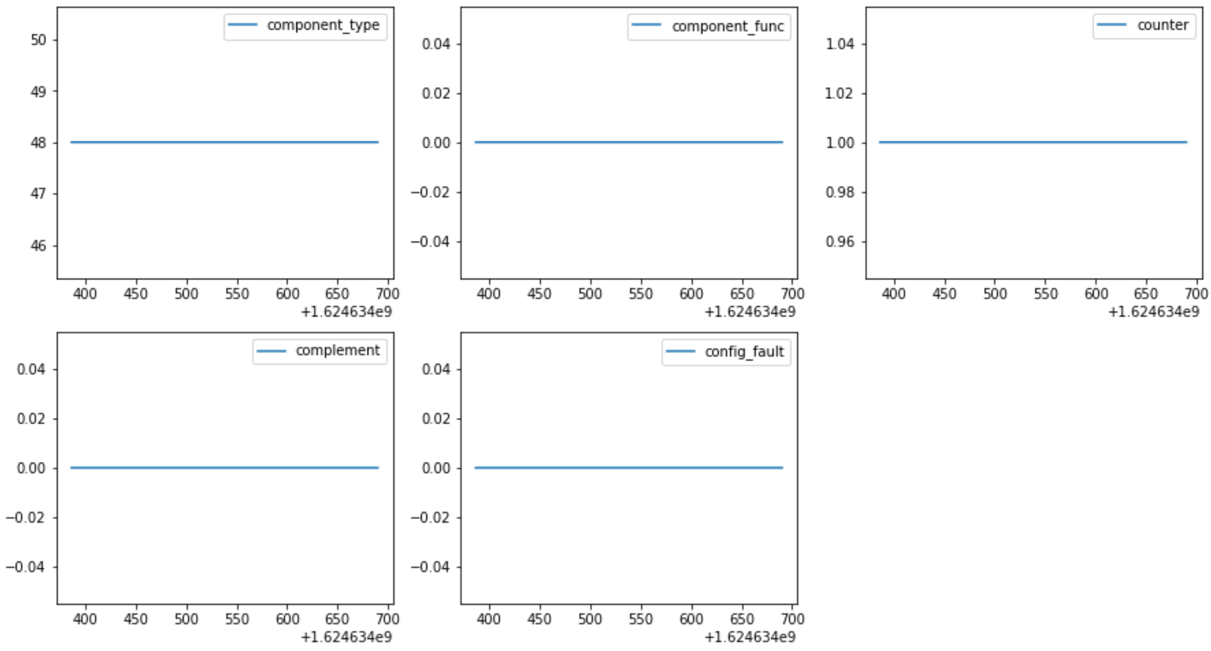
```
time = []
component_type = []
component_func = []
counter = []
complement = []
config_fault = []

for _, msg, t in b.read_messages('/pacmod/parsed_tx/component_rpt'):
    time.append(t.to_sec())

    component_type.append(msg.component_type)
    component_func.append(msg.component_func)
    counter.append(msg.counter)
    complement.append(msg.complement)
    config_fault.append(msg.config_fault)

print("====== Message example ======")
print(msg)

f = plt.figure(figsize=(15, 8))

ax1 = plt.subplot(2,3,1)
ax1.plot(time, component_type, label="component_type")
ax1.legend()

ax2 = plt.subplot(2,3,2)
ax2.plot(time, component_func, label="component_func")
ax2.legend()

ax3 = plt.subplot(2,3,3)
ax3.plot(time, counter, label="counter")
ax3.legend()

ax4 = plt.subplot(2,3,4)
ax4.plot(time, complement, label="complement")
```

```
ax4.legend()

ax5 = plt.subplot(2,3,5)
ax5.plot(time, config_fault, label="config_fault")
ax5.legend()
```

```
====== Message example ======
header:
  seq: 78140
  stamp:
    secs: 1624634689
    nsecs: 991403657
  frame_id: "pacmod"
component_type: 48
component_func: 0
counter: 1
complement: 0
config_fault: False
```

Out[20]:

```
<matplotlib.legend.Legend at 0x7f6d2994e7c0>
```



/pacmod/parsed_tx/global_rpt

In [21]:

```
for _, msg, t in b.read_messages('/pacmod/parsed_tx/global_rpt'):
    pass

print("====== Message example ======")
print(msg)
```

```
====== Message example ======
header:
  seq: 78141
  stamp:
    secs: 1624634689
    nsecs: 985570308
  frame_id: "pacmod"
enabled: True
override_active: False
fault_active: False
config_fault_active: False
user_can_timeout: False
brake_can_timeout: False
```

```
steering_can_timeout: False
vehicle_can_timeout: False
subsystem_can_timeout: False
user_can_read_errors: False
```

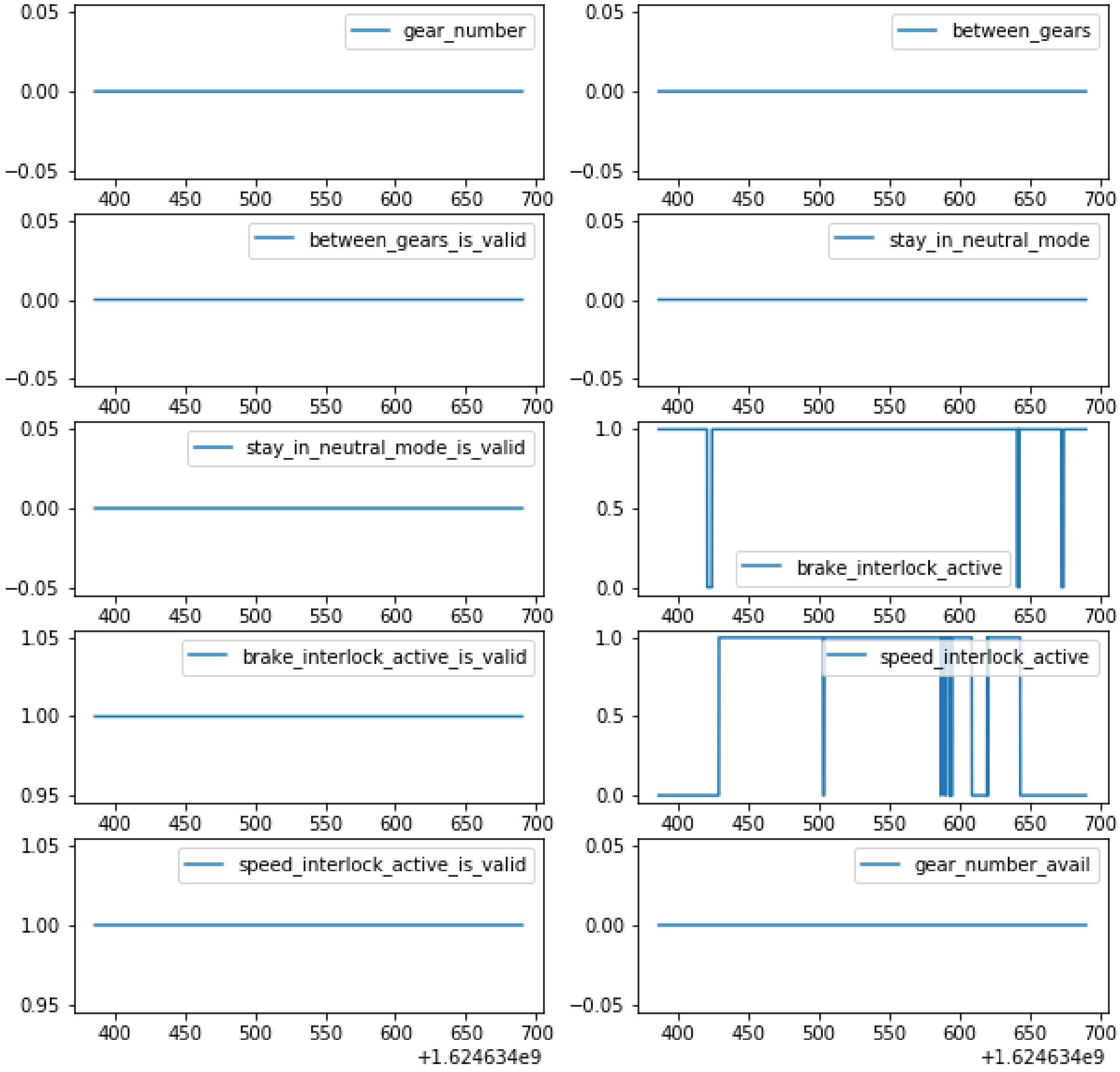
/pacmod/parsed_tx/shift_aux_rpt

In [22]:

```
time = []
gear_number = []

flags = {"between_gears": [],
        "between_gears_is_valid": [],
        "stay_in_neutral_mode": [],
        "stay_in_neutral_mode_is_valid": [],
        "brake_interlock_active": [],
        "brake_interlock_active_is_valid": [],
        "speed_interlock_active": [],
        "speed_interlock_active_is_valid": [],
        "gear_number_avail": []}

for _, msg, t in b.read_messages('/pacmod/parsed_tx/shift_aux_rpt'):
    time.append(t.to_sec())

    gear_number.append(msg.gear_number)

    flags["between_gears"].append(msg.between_gears)
    flags["between_gears_is_valid"].append(msg.between_gears_is_valid)
    flags["stay_in_neutral_mode"].append(msg.stay_in_neutral_mode)
    flags["stay_in_neutral_mode_is_valid"].append(msg.stay_in_neutral_mode_is_valid)
    flags["brake_interlock_active"].append(msg.brake_interlock_active)
    flags["brake_interlock_active_is_valid"].append(msg.brake_interlock_active_is_val
    flags["speed_interlock_active"].append(msg.speed_interlock_active)
    flags["speed_interlock_active_is_valid"].append(msg.speed_interlock_active_is_val
    flags["gear_number_avail"].append(msg.gear_number_avail)

print("====== Message example ======")
print(msg)

f = plt.figure(figsize=(10, 10))

ax1 = plt.subplot(5,2,1)
ax1.plot(time, gear_number, label="gear_number")
ax1.legend()

i = 2
for key, value in flags.items():
    ax_i = plt.subplot(5,2,i)
    ax_i.plot(time, value, label=key)
    ax_i.legend()
    i+=1
```

```
====== Message example ======
header:
  seq: 78159
  stamp:
    secs: 1624634689
    nsecs: 975699701
  frame_id: "pacmod"
between_gears: False
between_gears_is_valid: False
stay_in_neutral_mode: False
stay_in_neutral_mode_is_valid: False
```

```
brake_interlock_active: True
brake_interlock_active_is_valid: True
speed_interlock_active: False
speed_interlock_active_is_valid: True
gear_number_avail: False
gear_number: 0
```



/pacmod/parsed_tx/shift_rpt

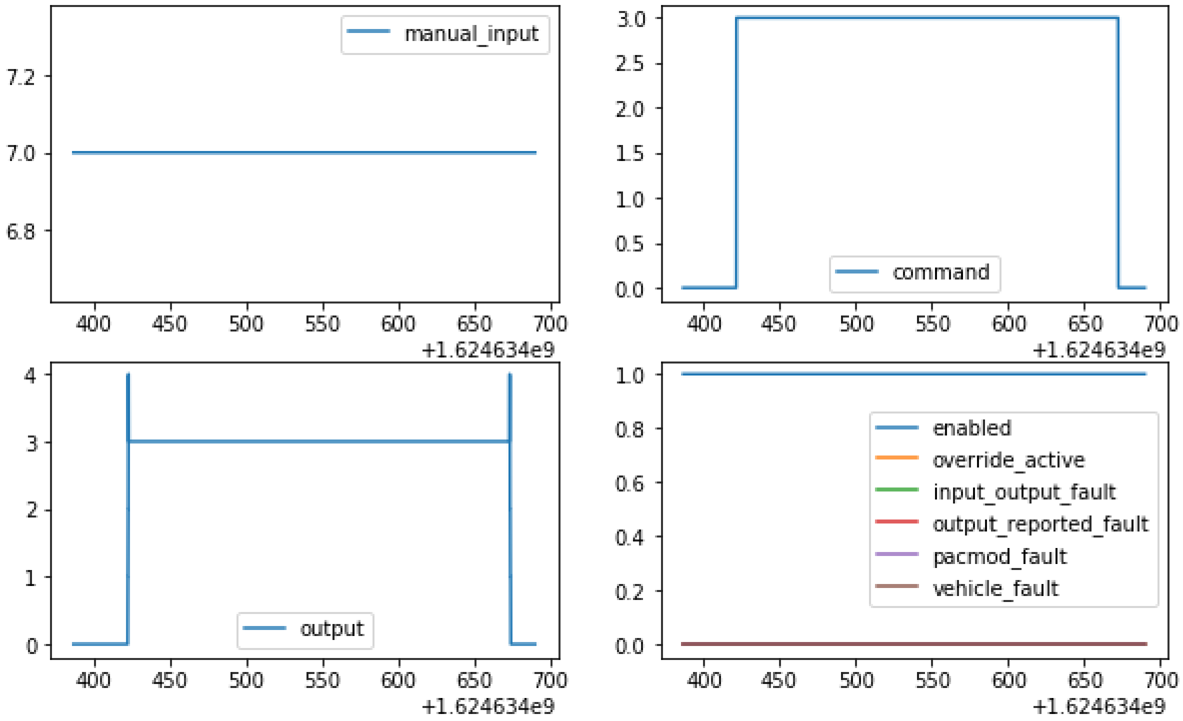
In [23]:

```
time = []
manual_input = []
command = []
output = []
flags = {"enabled": [],
        "override_active": [],
        "input_output_fault": [],
        "output_reported_fault": [],
        "pacmod_fault": [],
        "vehicle_fault": []}

for _, msg, t in b.read_messages('/pacmod/parsed_tx/shift_rpt'):
    time.append(t.to_sec())

    manual_input.append(msg.manual_input)
    command.append(msg.command)
    output.append(msg.output)
    flags["enabled"].append(msg.enabled)
    flags["override_active"].append(msg.override_active)
    flags["input_output_fault"].append(msg.input_output_fault)
    flags["output_reported_fault"].append(msg.output_reported_fault)
    flags["pacmod_fault"].append(msg.pacmod_fault)
    flags["vehicle_fault"].append(msg.vehicle_fault)

print("====== Message example ======")
print(msg)

f = plt.figure(figsize=(10, 6))

ax1 = plt.subplot(2,2,1)
ax1.plot(time, manual_input, label="manual_input")
ax1.legend()

ax2 = plt.subplot(2,2,2)
ax2.plot(time, command, label="command")
ax2.legend()

ax3 = plt.subplot(2,2,3)
ax3.plot(time, output, label="output")
ax3.legend()

ax4 = plt.subplot(2,2,4)
for key, value in flags.items():
    ax4.plot(time, value, label=key)
ax4.legend()
```

```
====== Message example ======
header:
  seq: 78160
  stamp:
    secs: 1624634690
    nsecs:   2554460
  frame_id: "pacmod"
enabled: True
override_active: False
command_output_fault: False
input_output_fault: False
output_reported_fault: False
pacmod_fault: False
vehicle_fault: False
manual_input: 7
command: 0
output: 0
```

Out[23]:

```
<matplotlib.legend.Legend at 0x7f6d29463dc0>
```



/pacmod/parsed_tx/steer_rpt

In [24]:

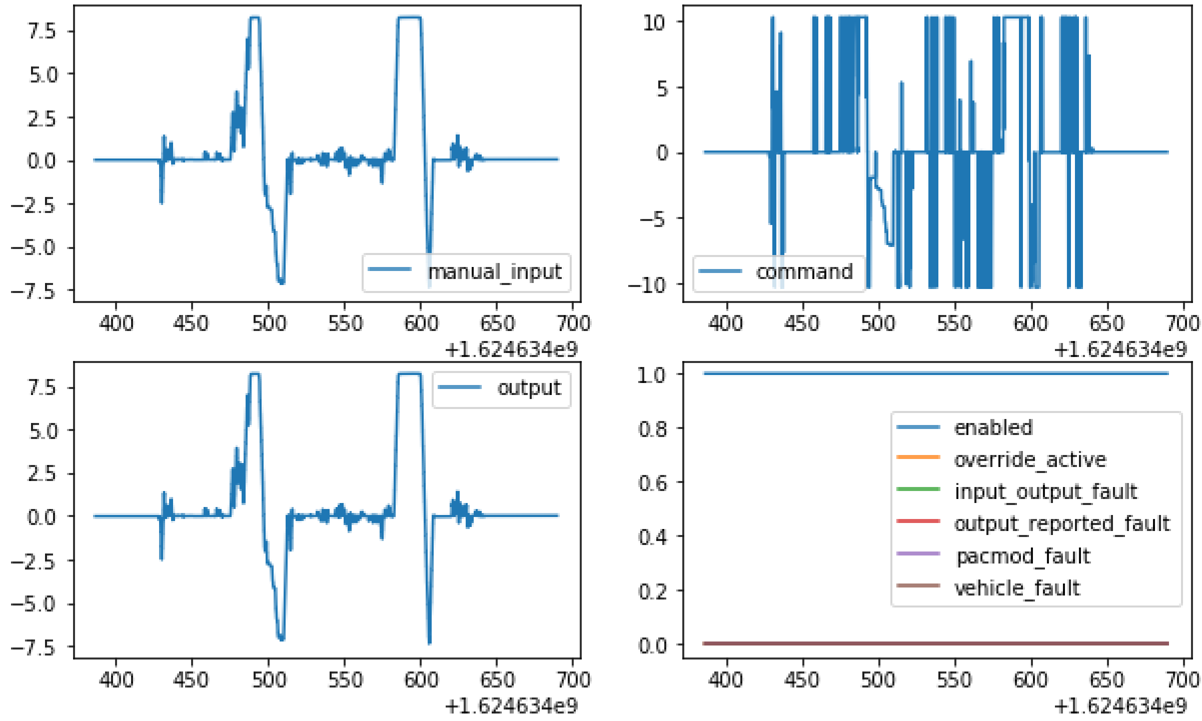
```
time = []
manual_input = []
command = []
output = []
flags = {"enabled": [],
        "override_active": [],
        "input_output_fault": [],
        "output_reported_fault": [],
        "pacmod_fault": [],
        "vehicle_fault": []}

for _, msg, t in b.read_messages('/pacmod/parsed_tx/steer_rpt'):
    time.append(t.to_sec())

    manual_input.append(msg.manual_input)
    command.append(msg.command)
    output.append(msg.output)

    flags["enabled"].append(msg.enabled)
    flags["override_active"].append(msg.override_active)
    flags["input_output_fault"].append(msg.input_output_fault)
    flags["output_reported_fault"].append(msg.output_reported_fault)
    flags["pacmod_fault"].append(msg.pacmod_fault)
    flags["vehicle_fault"].append(msg.vehicle_fault)

print("====== Message example ======")
print(msg)

f = plt.figure(figsize=(10, 6))

ax1 = plt.subplot(2,2,1)
ax1.plot(time, manual_input, label="manual_input")
ax1.legend()

ax2 = plt.subplot(2,2,2)
ax2.plot(time, command, label="command")
```

```
ax2.legend()

ax3 = plt.subplot(2,2,3)
ax3.plot(time, output, label="output")
ax3.legend()

ax4 = plt.subplot(2,2,4)
for key, value in flags.items():
    ax4.plot(time, value, label=key)
ax4.legend()
```

```
====== Message example ======
header:
  seq: 78132
  stamp:
    secs: 1624634689
    nsecs: 980942138
  frame_id: "pacmod"
enabled: True
override_active: False
command_output_fault: False
input_output_fault: False
output_reported_fault: False
pacmod_fault: False
vehicle_fault: False
manual_input: 0.017
command: 0.0
output: 0.017
```

Out[24]:

```
<matplotlib.legend.Legend at 0x7f6d292a6520>
```



/pacmod/parsed_tx/steer_rpt_detail_1

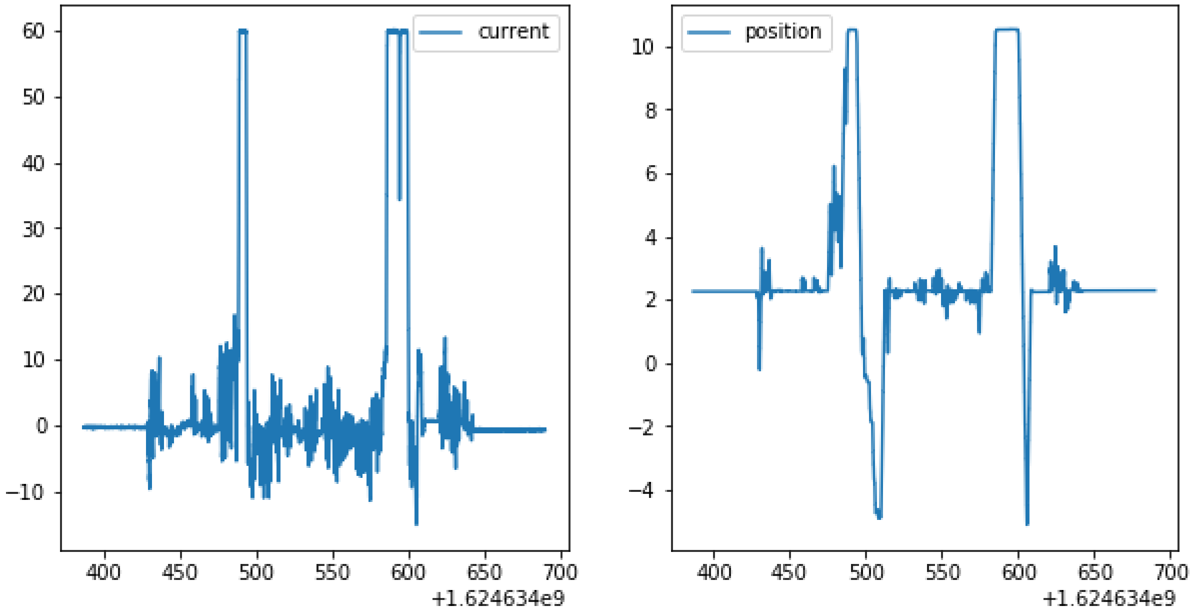
In [25]:

```
time = []
current = []
position = []

for _, msg, t in b.read_messages('/pacmod/parsed_tx/steer_rpt_detail_1'):
    time.append(t.to_sec())
    current.append(msg.current)
    position.append(msg.position)

print("====== Message example ======")
print(msg)

f = plt.figure(figsize=(10, 5))

ax1 = plt.subplot(1,2,1)
ax1.plot(time, current, label="current")
ax1.legend()

ax2 = plt.subplot(1,2,2)
ax2.plot(time, position, label="position")
ax2.legend()
```

```
====== Message example ======
header:
  seq: 78118
  stamp:
    secs: 1624634689
    nsecs: 989526979
  frame_id: "pacmod"
current: -0.81
position: 2.297
```

Out[25]:

```
<matplotlib.legend.Legend at 0x7f6d29294340>
```



/pacmod/parsed_tx/steer_rpt_detail_2

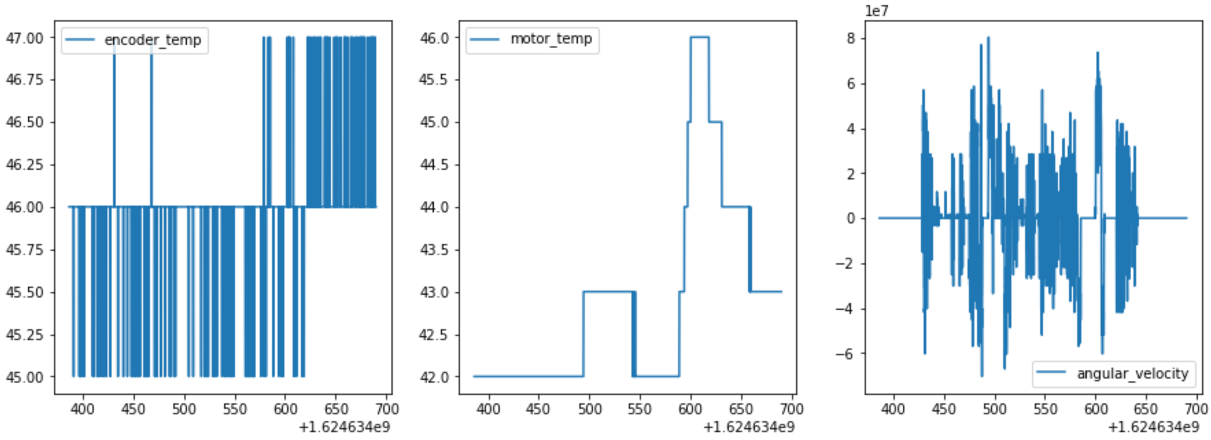
In [26]:

```
time = []
encoder_temp = []
motor_temp = []
angular_velocity = []

for _, msg, t in b.read_messages('/pacmod/parsed_tx/steer_rpt_detail_2'):
    time.append(t.to_sec())

    encoder_temp.append(msg.encoder_temp)
    motor_temp.append(msg.motor_temp)
    angular_velocity.append(msg.angular_velocity)

print("====== Message example ======")
print(msg)

f = plt.figure(figsize=(15, 5))

ax1 = plt.subplot(1,3,1)
ax1.plot(time, encoder_temp, label="encoder_temp")
ax1.legend()

ax2 = plt.subplot(1,3,2)
ax2.plot(time, motor_temp, label="motor_temp")
ax2.legend()

ax3 = plt.subplot(1,3,3)
ax3.plot(time, angular_velocity, label="angular_velocity")
ax3.legend()
```

```
====== Message example ======
header:
  seq: 78115
  stamp:
    secs: 1624634690
    nsecs:   5458471
  frame_id: "pacmod"
encoder_temp: 46.0
motor_temp: 43.0
angular_velocity: 0.0
```

Out[26]:

```
<matplotlib.legend.Legend at 0x7f6d291815e0>
```



/pacmod/parsed_tx/steer_rpt_detail_3

In [27]:

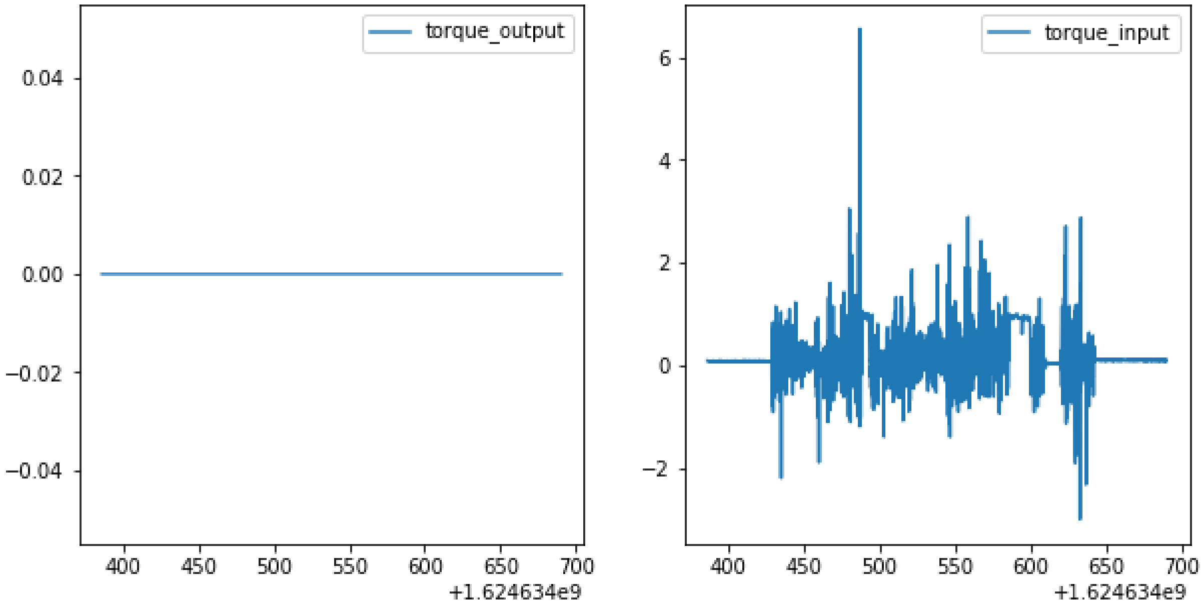
```
time = []
torque_output = []
torque_input = []

for _, msg, t in b.read_messages('/pacmod/parsed_tx/steer_rpt_detail_3'):
    time.append(t.to_sec())

    torque_output.append(msg.torque_output)
    torque_input.append(msg.torque_input)

print("====== Message example ======")
print(msg)

f = plt.figure(figsize=(10, 5))

ax1 = plt.subplot(1,2,1)
ax1.plot(time, torque_output, label="torque_output")
```

```
ax1.legend()

ax2 = plt.subplot(1,2,2)
ax2.plot(time, torque_input, label="torque_input")
ax2.legend()
```

```
====== Message example ======
header:
  seq: 78108
  stamp:
    secs: 1624634689
    nsecs: 974311449
  frame_id: "pacmod"
torque_output: 0.0
torque_input: 0.098
```

Out[27]:

```
<matplotlib.legend.Legend at 0x7f6d2905dbb0>
```



/pacmod/parsed_tx/vehicle_speed_rpt

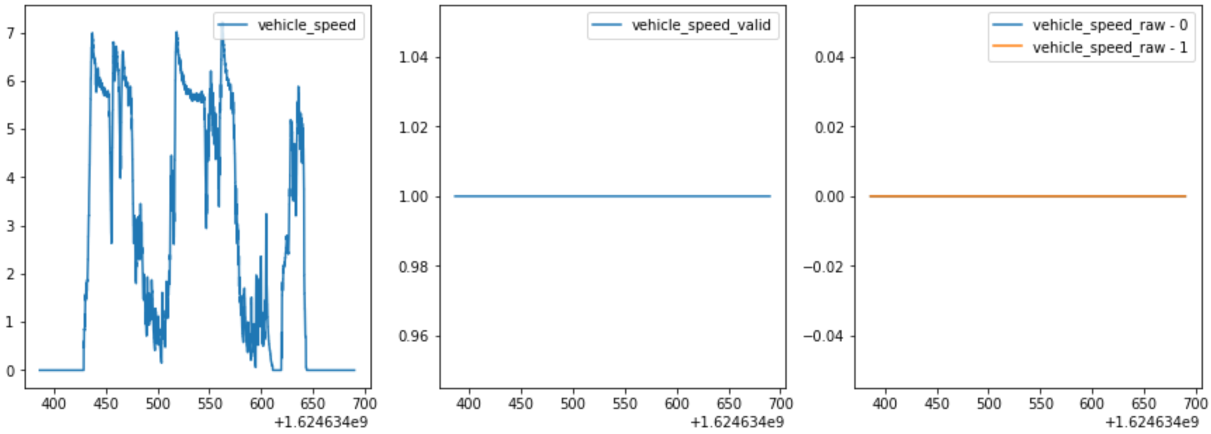
In [28]:

```
time = []
vehicle_speed = []
vehicle_speed_valid = []
vehicle_speed_raw = []

for _, msg, t in b.read_messages('/pacmod/parsed_tx/vehicle_speed_rpt'):
    time.append(t.to_sec())
    vehicle_speed.append(msg.vehicle_speed)
    vehicle_speed_valid.append(msg.vehicle_speed_valid)
    vehicle_speed_raw.append([msg.vehicle_speed_raw[0], msg.vehicle_speed_raw[0]])

print("====== Message example ======")
print(msg)
print("vehicle_speed_raw is with type:", type(msg.vehicle_speed_raw))

vehicle_speed_raw = np.array(vehicle_speed_raw)

f = plt.figure(figsize=(15, 5))

ax1 = plt.subplot(1,3,1)
ax1.plot(time, vehicle_speed, label="vehicle_speed")
ax1.legend()
```

```
ax2 = plt.subplot(1,3,2)
ax2.plot(time, vehicle_speed_valid, label="vehicle_speed_valid")
ax2.legend()

ax3 = plt.subplot(1,3,3)
ax3.plot(time, vehicle_speed_raw[:, 0], label="vehicle_speed_raw - 0")
ax3.plot(time, vehicle_speed_raw[:, 1], label="vehicle_speed_raw - 1")
ax3.legend()
```

```
====== Message example ======
header:
  seq: 78130
  stamp:
    secs: 1624634689
    nsecs: 985869600
  frame_id: "pacmod"
vehicle_speed: 0.0
vehicle_speed_valid: True
vehicle_speed_raw: [0, 0]
vehicle_speed_raw is with type: <class 'bytes'>
```

```
Out[28]: <matplotlib.legend.Legend at 0x7f6d28fdf3a0>
```


vehicle_speed
+1.624634e9
vehicle_speed_valid
+1.624634e9
vehicle_speed_raw - 0
vehicle_speed_raw - 1
+1.624634e9