```
#!/usr/bin/env python

import rospy
from turtlesim.msg import Pose
from geometry_msgs.msg import Twist
from std_msgs.msg import Bool
from math import sqrt

# Variables globales
pose = Pose()
waypoint = (7, 7)

# Callback de la souscription au topic "pose"
def pose_callback(data):
    global pose
    pose = data

# Calcul de la distance euclidienne entre deux points
def calculate_distance(point1, point2):
    return sqrt((point2[1] - point1[1])**2 + (point2[0] - point1[0])**2)

# Fonction principale
def set_way_point():
    # Initialisation du nœud ROS
    rospy.init_node('set_way_point', anonymous=True)

    # Souscription au topic "pose"
    rospy.Subscriber("pose", Pose, pose_callback)

    # Création du publisher pour cmd_vel
    cmd_vel_pub = rospy.Publisher('cmd_vel', Twist, queue_size=10)
```

```
# Création du publisher pour is_moving
is_moving_pub = rospy.Publisher('is_moving', Bool, queue_size=10)

# Paramètre Kpl (constante de régulation en distance)
kpl = rospy.get_param('~Kpl', 1.0)

# Paramètre distance_tolerance
distance_tolerance = rospy.get_param('~distance_tolerance', 0.1)

# Taux de rafraîchissement
rate = rospy.Rate(10)  # 10Hz

while not rospy.is_shutdown():
    # Calcul de la distance entre le waypoint et la position de la tortue
    distance = calculate_distance((pose.x, pose.y), waypoint)

    if distance > distance_tolerance:
        # Calcul de l'erreur linéaire
        error_linear = distance

        # Commande linéaire
        v = kpl * error_linear

        # Création du message Twist pour la commande de vitesse linéaire
        twist_msg = Twist()
        twist_msg.linear.x = v

        # Publication du message Twist
        cmd_vel_pub.publish(twist_msg)
```

```
            # Publication de True sur le topic is_moving
            is_moving_pub.publish(True)
        else:
            # Publication de False sur le topic is_moving
            is_moving_pub.publish(False)

        rate.sleep()

if __name__ == '__main__':
    try:
        set_way_point()
    except rospy.ROSInterruptException:
        pass
```